

量子群智能及其在通信技术中的应用

高洪元 刁 鸣 著

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书共分9章, 主要内容包括量子粒子群、量子蛙跳算法、量子蜂群、量子细菌觅食、量子神经网络、量子文化算法、量子文化群智能、量子膜群算法、多目标量子膜群算法、多用户检测、频谱分配、决策引擎、频谱感知、冲击噪声测向、非圆信号测向等智能计算或通信技术中的热点和难点问题。

本书可使读者在了解和学习量子群智能和通信技术最新科研成果的同时, 在量子群智能和通信技术两个方向得到启发, 也可作为相关学科的教材和科研用书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目(CIP)数据

量子群智能及其在通信技术中的应用 / 高洪元, 刁鸣著. —北京: 电子工业出版社, 2016.6

ISBN 978-7-121-28814-2

I. ①量… II. ①高… ②刁… III. ①量子群—智能技术—应用—通信技术 IV. ①TN91

中国版本图书馆 CIP 数据核字 (2016) 第 101124 号

策划编辑: 董亚峰

责任编辑: 董亚峰

特约编辑: 田学清 赵海红

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 720×1000 1/16 印张: 15.25 字数: 304 千字

版 次: 2016 年 6 月第 1 版

印 次: 2016 年 6 月第 1 次印刷

定 价: 48.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件到 dbqq@phei.com.cn。

本书咨询联系方式: QQ3502629。

前言

无线通信是当今科学技术发展最活跃的领域。在无线通信飞速发展的今天，频谱资源和系统性能都难以跟上实际需求的快速增长，成为制约无线通信持续发展的主要因素。研究通信系统的关键技术对合理分配频谱资源、确定系统参数和满足系统多目标要求有深远意义。因此，本书根据智能计算的最新进展，在量子群智能理论的基础上进行学科交叉，介绍了量子群智能频谱分配、频谱感知、多用户检测、测向和决策引擎等具有重要的理论价值和现实意义的通信关键技术。基于量子群智能的通信关键技术是当前智能计算和通信领域的研究热点，在工业和国防科技领域具有广阔的应用前景。希望本书的出版将对量子群智能和通信技术的持续发展起到推进作用。

本书阐述了作者及其团队在量子群智能领域的学术研究成果及其在通信技术中的应用，阐明了在通信技术中使用量子群智能的可行性、有效性及其对后续科研思路的启发。量子群智能在通信技术中的应用是智能计算与通信技术领域结合的一个前沿和富有挑战性的研究方向，它以量子群智能理论为基础，侧重于介绍如何设计目标函数去解决通信技术中所面对的技术难题，如何根据工程问题设计量子群智能算法去快速可靠地解决工程优化问题。本书在介绍量子智能相关理论的基础上，阐述了智能计算的新发展——量子群智能计算，内容主要包括：量子粒子群算法、量子蛙跳算法、量子蜂群算法、量子细菌觅食算法、量子神经网络、量子文化算法、量子文化群智能、量子膜群算法和多目标量子膜群算法。并且给出了量子群智能在通信技术中的具体应用，解决了多用户检测、频谱分配、决策引擎、频谱感知、冲击噪声测向和非圆信号测向等通信技术中的热点和难点问题。同时介绍了当前量子群智能及其应用的一些最新技术，如多目标频谱分配、量子膜群、量子文化群、智能非圆测向和智能冲击

噪声环境测向等通信技术和智能计算领域的热点问题，做到理论和具体应用的有机结合。

本书内容安排尽可能考虑适合高等学校相关专业研究生及博士生的实际研究和教学要求，做到深入浅出、重点突出，读者可以在相对较短的时间内入门并深入进去。本书可使读者在了解和学习量子群智能和通信技术最新科研成果的同时，在量子群智能和通信技术两个方向得到启发。本书可作为相关学科研究人员的教学和科研参考用书。

本书第 4、8、9 章的部分内容由刁鸣教授撰写，其他章节内容由高洪元教授撰写。在著作写作过程中，感谢硕士研究生李晨琬、杜亚男、李佳、刘丹丹和梁炎松在文献和文稿整理方面所付出的工作和努力。由于量子群智能和通信技术发展迅猛、应用广泛，再加上作者的水平有限，难免存在缺点和不足之处，敬请读者批评指正！尽管本书主要探讨量子群智能算法在通信技术中的应用，但是书中的一些量子群智能算法可以移植应用到其他工程领域，本着相互学习、相互促进的初衷，欢迎读者使用如下邮箱进行沟通交流：gaohongyuan@hrbeu.edu.cn。

本专著获得了如下基金的资助：国家自然科学基金（No.61102106, No.61571149）、中国博士后科学基金特别资助（No.2015T80325）、中国博士后科学基金 1 等资助（No.2013M530148）、黑龙江省博士后科学基金（No.LBHZ13054）和中央高校基本科研业务费（No.HEUCF150817），在此表示衷心感谢。本专著的撰写主要在访学期间完成，特此感谢中国留学基金委的资助。

高洪元

目 录

第 1 章 绪论.....	1
1.1 量子群智能计算简介	1
1.2 单目标和多目标优化问题	3
1.2.1 单目标优化问题的数学模型	5
1.2.2 单目标约束优化问题的求解方法	5
1.2.3 多目标优化问题的数学模型	7
1.2.4 多目标优化方法简介	9
1.3 智能计算在通信技术中的应用	11
1.3.1 简介	11
1.3.2 展望	15
1.4 本书内容和结构安排	17
参考文献	19
第 2 章 用于离散优化问题的量子群智能计算.....	29
2.1 量子粒子群算法	30
2.1.1 双链编码的量子粒子群算法	30
2.1.2 单链编码的量子粒子群算法	32
2.1.3 性能测试	34
2.2 量子蜂群算法	37
2.2.1 双链编码的量子蜂群算法	37
2.2.2 单链编码的量子蜂群算法	40
2.2.3 性能测试	42
2.3 量子细菌觅食算法	43

2.3.1	双链编码的量子细菌觅食算法	43
2.3.2	量子细菌觅食算法的收敛性分析	46
2.3.3	性能测试	49
2.4	小结	50
	参考文献	50
第 3 章	用于连续优化问题的量子群智能计算	52
3.1	量子蛙跳算法	53
3.1.1	混合蛙跳算法	53
3.1.2	量子蛙跳算法	55
3.1.3	性能测试	58
3.2	量子文化蛙跳算法	60
3.2.1	量子规范知识	60
3.2.2	量子文化蛙跳算法的实现	61
3.2.3	性能测试	64
3.3	量子细菌觅食算法	65
3.3.1	细菌觅食算法	66
3.3.2	量子细菌觅食优化算法	69
3.3.3	性能测试	71
3.4	小结	73
	参考文献	74
第 4 章	基于量子智能算法的多用户检测	76
4.1	多用户检测的数学模型	78
4.1.1	噪声模型	78
4.1.2	高斯噪声下的多用户检测数学模型	79
4.2	高斯噪声环境的典型多用户检测方法	83
4.2.1	传统检测器	83

第 5 章	基于量子群智能的认知无线电决策引擎	112
5.1	认知无线电决策引擎模型和三种典型的决策引擎	114
5.1.1	智能计算的认知决策引擎	114
5.1.2	基于智能计算的认知决策引擎方法	117
5.2	单目标膜量子蜂群算法及其在决策引擎上的应用	123
5.2.1	膜结构简介	123
5.2.2	膜量子蜂群优化算法	124
5.2.3	膜量子蜂群优化算法的性能测试	128
5.2.4	基于膜量子蜂群算法的认知无线电决策引擎	130
5.2.5	决策引擎试验仿真	131

5.3 基于量子细菌觅食算法的绿色认知无线电参数调整	134
5.3.1 绿色认知无线电参数调整模型	135
5.3.2 量子细菌觅食算法的绿色认知无线电参数调整	137
5.3.3 试验仿真	138
5.4 小结	141
参考文献	142
第6章 基于量子群智能的频谱分配	146
6.1 频谱分配模型	147
6.1.1 图论着色模型	147
6.1.2 单目标频谱分配	149
6.1.3 多目标频谱分配	150
6.2 基于量子粒子群算法的单目标频谱分配	151
6.2.1 基于单链量子粒子群算法的认知无线电频谱分配	151
6.2.2 仿真结果分析	152
6.3 基于多目标膜量子蜂群的多目标频谱分配	155
6.3.1 膜量子蜂群的基本演进规则	156
6.3.2 膜量子蜂群的膜框架	158
6.3.3 基于多目标膜量子蜂群算法的多目标频谱分配技术	161
6.3.4 频谱分配实验仿真	163
6.4 小结	167
参考文献	168
第7章 量子群智能的频谱感知技术	171
7.1 线性协作频谱感知模型	172
7.2 合作式频谱感知的基本算法	174
7.2.1 基于修正偏差因子的频谱感知方法	174
7.2.2 基于粒子群算法的频谱感知	175

7.2.3 计算机仿真	177
7.3 基于连续量子细菌觅食算法的频谱感知技术	178
7.3.1 量子细菌觅食算法的频谱感知	178
7.3.2 计算机仿真	181
7.4 小结	184
参考文献	184
第 8 章 基于量子智能计算的 DOA 估计	187
8.1 经典 DOA 估计模型和算法	188
8.1.1 DOA 估计模型	188
8.1.2 经典测向算法	189
8.1.3 基于量子蛙跳算法的测向方法	190
8.1.4 试验仿真	191
8.2 基于高阶累积量和文化量子算法的测向方法	192
8.2.1 基于高阶累积量的广义加权子空间拟合算法	193
8.2.2 文化量子算法	196
8.2.3 基于文化量子算法的广义高阶加权信号子空间拟合测向	199
8.2.4 试验仿真	200
8.3 基于量子文化蛙跳算法的非圆信号 DOA 估计	203
8.3.1 非圆极大似然算法	203
8.3.2 基于量子文化蛙跳算法的极大似然测向	204
8.3.3 试验仿真	206
8.4 小结	207
参考文献	208
第 9 章 冲击噪声环境下的量子智能计算 DOA 估计	211
9.1 冲击噪声环境下的测向模型	212
9.1.1 三种低阶矩	212

9.1.2	三种低阶矩的对比	213
9.2	基于量子文化细菌觅食算法的无穷范数最大似然测向方法.....	215
9.2.1	量子文化细菌觅食算法	215
9.2.2	量子文化细菌觅食算法的无穷范数极大似然测向.....	218
9.2.3	试验仿真	221
9.3	基于量子粒子群的动态测向方法	225
9.3.1	冲击噪声下的动态测向模型	225
9.3.2	连续量子粒子群优化算法	226
9.3.3	量子粒子群优化算法的动态测向方法.....	228
9.3.4	试验仿真	229
9.4	小结	231
	参考文献	232

第 1 章

绪 论

1.1 量子群智能计算简介

随着科学技术的不断发展和学科间的互相交叉渗透，许多领域的研究问题和方法也变得越来越复杂，许多课题也不再局限于用一个领域的理论和方法去解决，这在信息科学领域尤为突出。面对复杂的研究问题，当传统方法在工程应用中陷入困境的时候，其促使人们不断从自然界寻找灵感，更加艺术地寻求解决问题的新方法和新手段。正是在这样的背景下，智能计算在理论研究和工程应用领域得到重视并逐渐发展起来。

长久以来，科学家通过向自然界学习并从中获得灵感，提出新的计算理论，促进了启发式计算的兴起，涌现出诸多受生物启发的计算方法，如神经网络、禁忌搜索、演进算法和群智能等。这些方法通过模拟生物的行为或物理现象进而抽象出数学模型，去解决理论计算和工程领域中遇到的问题。随着人类文明的进步，物理科学、生物科学及信息科学的进一步发展，科学家对生物智能和启发式算法的认识不断深入，越来越多的研究人员认识到对生物系统行为和社会行为的模拟可以获得解决众多复杂问题的方法。受自然界和人类社会活动的启发不断涌现出新的智能计算方法，如利用 DNA 计算、细胞膜计算、量子计

算、文化算法、群集智能和人工免疫系统等,将这些计算方法统称为智能计算(Intelligence Computing, IC)。实际上,智能计算是一门包括物理学、数学、计算机科学、电子、信息、通信、生理学、社会学、进化理论和心理学等学科在内的综合科学^[1]。因此,将各研究领域的经典理论和智能计算理论相结合,提出的理论就能为解决工程问题建立有效的智能系统,并可为理论研究提供新方法,对各个科学领域的发展起到推进作用。

当前,一种被称作群智能的典型智能计算算法受到科研人员的广泛关注,这种群智能的优化算法也可被称为启发式算法,它是受生物群体性行为的启发而设计提出的。这类算法于在 20 世纪 90 年代兴起,被广泛应用于解决复杂的组合优化问题及高维优化问题。由于大部分群智能算法的演进过程是独立于初始解并自由演化的,所以它能够克服传统优化算法及确定性优化方法的一些缺点,如陷入局部最优值或者获得边缘化解等。因此,一些学者把研究焦点放在了群智能算法上,如粒子群算法^[2,3]、人工蜂群算法^[4]、细菌觅食算法^[5]和混合蛙跳算法^[6]等,这些算法通过对群体生物的群体性觅食行为进行模拟而具有较好的寻优能力,因而被广泛应用于许多工程领域。这些算法具有不同的特点,在解决一些优化问题时能够获得令人满意的解。但是在求解一些复杂优化问题时会有收敛速度与收敛精度之间的矛盾,在求解高维问题时,需要较大的种群规模和较多的迭代次数,这无疑增加了计算量及运行时间。因此,设计高效、稳定且适用性广的算法是十分必要的。

量子计算智能是受物理学中量子计算和量子机制的启发所产生的智能优化算法。量子计算(Quantum Computation, QC)于 1982 年首先由 Richard Feynman 提出,现已成为各国学者紧密关注的前沿学科^[7]。量子计算已经在保密通信、密码系统和数据库搜索等领域得到成功的应用^[8,9]。虽然在 1999 年美国已经研制出量子计算设备的原型,但其在实际应用上和技术上仍然是一个巨大的挑战,然而关于量子计算令人激动的原因是它引起思考物理学基本定律所获得的心得,以及它为其他科学的发展所带来的独创性的思想和灵感^[10]。

量子计算生动地说明了智能计算的研究也可以建立在一个物理过程的模拟之上,量子机理和特性会为智能计算的研究开辟新的途径,有效利用量子理论

的原理和概念，将会在实际应用中取得明显优于传统智能计算模型的结果^[11]，这是因为量子计算智能（Quantum Computational Intelligence, QCI）的出现结合了量子计算和传统智能计算各自的优势^[12,13]。量子计算与进化算法的结合从 20 世纪 90 年代后期开始。Han 等提出了遗传量子算法^[14]和量子衍生进化算法^[15]，他们将量子编码引入遗传进化过程，利用量子旋转门实现量子染色体的进化，并且给出了量子旋转角的确定方法，并以量子旋转门进化，通过量子态测量实现了 0/1 背包问题的求解，其性能与传统的遗传算法相比有优势。此后，由于量子计算的并行性特点，使量子进化算法具有很多优点，如种群规模小也不影响算法性能、迭代次数少且全局寻优能力强、在进化过程中有效利用了个体过去的历史信息等，因此吸引了众多研究者对量子进化算法进行改进和完善，并在工程问题上进行了应用研究的探索，并用实验结果证明其性能更加良好^[16-20]。文献[21,22]通过对具体工程问题的研究并结合一些其他的仿生机制对算法进行改进，获得了较理想的性能。这类量子启发式算法与传统进化算法相比，搜索能力显著增强，且运算量和运算时间都极大地缩减，应用在各领域的工程问题上都能获得较为理想的性能。因此，对量子计算及群智能进化算法进行深入研究，设计出全新的量子群智能算法具有重要的意义^[23-25]。深入研究量子机理，结合最新的智能计算成果同时使用量子旋转门实现的演化过程设计出量子编码的演化计算新方法，将具有很高的理论价值和发展潜力^[26,27]。

1.2 单目标和多目标优化问题

对于优化问题来说，一般主要由目标函数、未知变量和约束条件 3 个基本要素构成。它们的概念详细阐述如下^[28]。

（1）目标函数：就是要求解的优化问题的函数表示，即要求被最大化或者最小化的函数。

(2) 未知变量：就是目标函数中未知的变量，通过自身取值变化而引起目标函数值变化的变量，求解优化问题的目标也就是希望能够找到满足目标函数最大化或者最小化所对应的最优变量取值。

(3) 约束条件：就是在寻找使得目标函数最大化或最小化的未知变量时，限制未知变量取值条件的约束。基本上每个优化问题都要定义一组边界约束，并给出每个变量的定义域范围。

所以一个优化问题的求解过程就可以理解为：在满足给定约束条件的未知变量的可行区域中，通过一定的方式和方法寻找出那组能够使得目标函数达到最优的变量取值，并由此得出目标函数的最优值。当然这里的最优视具体情况而定，可以是函数的最大值，也可以是函数的最小值。那么，优化算法就是作为优化问题程序化时的求解方法，在指定约束条件的可选解集空间中搜索满足函数最大化或者最小化的最优解。

以上是对优化问题的基本概念介绍，现在基于优化问题的 3 个组成要素的不同特征，分别介绍优化问题最常见的 3 种分类方法。

(1) 从目标函数的维度来进行分类，根据目标函数的个数可以分为单目标优化问题和多目标优化问题。单目标优化问题顾名思义就是指优化问题中的目标函数只有一个，而如果目标函数有两个以上则称之为多目标优化问题。

(2) 根据未知变量类型的不同可以分为连续域优化问题和离散域优化问题。所谓连续域优化问题就是指未知变量的定义域取值为连续域的，同理离散域优化问题就是指未知变量的定义域取值为离散域的。

(3) 根据有无约束条件的情况可分为无约束条件的优化问题和有约束条件的优化问题。

接下来以最小值优化为例介绍单目标优化问题和多目标优化问题，同时了解优化问题的基本术语。最大值优化问题可依此类推。

1.2.1 单目标优化问题的数学模型

1. 单目标函数优化问题的数学描述

令 $S \subseteq R^n$ 代表 n 维的搜索空间, 令 $\mathbf{x} = [x_1, x_2, \dots, x_n] \in S$ 表示一个决策向量, $x_i \in [x_i^{\min}, x_i^{\max}]$ 代表边界约束。令 $f(\mathbf{x})$ 为目标函数, 以最小化为例, 则单目标优化问题可以描述为

$$\min f(\mathbf{x}), x_i \in [x_i^{\min}, x_i^{\max}], i = 1, 2, \dots, n \quad (1.2.1)$$

2. 单目标约束优化问题

在科学与工程领域中, 许多极值问题的求解往往受到各种现实因素的制约, 这些制约通常由一系列约束条件来描述。求解带有约束条件的极值问题被称为约束优化问题, 具体可由下述一般形式的非线性规划来表示:

$$\min_{\mathbf{x} \in S} f(\mathbf{x}), \text{st} : g_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, s; h_j(\mathbf{x}) = 0, j = s+1, s+2, \dots, p \quad (1.2.2)$$

其中, $\mathbf{x} = [x_1, x_2, \dots, x_n]$ 是 n 维实向量, $f(\mathbf{x})$ 为目标函数, $g_i(\mathbf{x})$ 表示第 i 个不等式约束, $h_j(\mathbf{x})$ 表示第 j 个等式约束, 决策变量 x_i 在区间 $[x_i^{\min}, x_i^{\max}]$ 中取值, $i = 1, 2, \dots, n$ 。 $S = \prod_{i=1}^n [x_i^{\min}, x_i^{\max}]$ 表示搜索空间, S 中所有满足约束条件的可行解构成的可行域记为 $\Omega \subseteq S$ 。

1.2.2 单目标约束优化问题的求解方法

由于约束条件的存在, 使得约束极值问题的求解要比无约束优化问题的求解复杂、困难得多。对于约束极小化问题来说, 不仅要使目标函数值在迭代过程中不断减小, 而且还要注意解的可行性。为了简化约束优化问题的寻优过程, 通常可采用如下思路来构造算法: 将约束优化问题转化为无约束优化问题, 将复杂问题转化为简单问题。下面对 3 种代表性方法进行简单介绍。

1. 基于罚函数的方法

(1) 静态罚函数法：考虑如下非线性规划问题

$$\min_{\mathbf{x} \in S} f(\mathbf{x}), \text{st} : g_i(\mathbf{x}) \leq 0, i=1, 2, \dots, m \quad (1.2.3)$$

适应度函数定义为： $F(\mathbf{x}) = f(\mathbf{x}) + p(\mathbf{x})$ 。

惩罚项 $p(\mathbf{x})$ 定义为：若 \mathbf{x} 可行，则 $p(\mathbf{x}) = 0$ ；否则 $p(\mathbf{x}) = \sum_{i=1}^m r_i g_i(\mathbf{x})$ 。由

于对于每个约束又可分为几个违反级，所以按照违反级， r_i 是相应变化的约束 i 的惩罚系数。

(2) 动态罚函数法：该方法使用了与时间（迭代次数、代数等） t 有关的罚函数，在第 t 代个体的适应度函数设置为

$$F(\mathbf{x}) = f(\mathbf{x}) + (ct)^\alpha \sum_{i=1}^m d_i^\beta(\mathbf{x}) \quad (1.2.4)$$

其中， c, α, β 为可调参数， $d_i(\mathbf{x})$ 在 \mathbf{x} 为可行解时取 0，在 \mathbf{x} 为不可行解时取约束违反量的绝对值。

2. 基于搜索容许解的方法

如行为记忆法，该方法逐步处理约束，每次处理一个，使得种群中满足该约束的个体达到预先规定的比例，如在处理第 j 个约束的时候，要保持前 $j-1$ 个约束均满足，这样使得最后种群中有足够多的可行解。相关的方法还有容许点优先法等。

3. 多目标方法

多目标方法的特点是：既不使用传统的罚函数，也不区分可行解和不可行解。有的算法将约束优化问题转化为两个目标优化问题，其中一个为原问题的目标函数，另一个为违反约束条件的程度函数。利用 Pareto 优于关系，定义个体 Pareto 序值以便对个体进行排序选优。有的算法采用序值阶段选择策略，第一阶段，当种群中没有可行解时，按照约束违反度对种群中的个体进行排序，

选择约束违反度较小的个体组成下一代种群；若种群中的不可行解个数不为 0，则采用第二阶段选择策略。

1.2.3 多目标优化问题的数学模型

多目标优化问题（MOP）又称为多标准优化问题、多性能优化问题或者多矢量优化问题^[29]。不失一般性，一个具有 n 个决策变量、 m 个目标函数的多目标优化问题可表述为

$$\begin{aligned} \min \{ & \mathbf{y} = \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})] \} \\ \text{s.t. : } & g_i(\mathbf{x}) \leq 0, i=1, 2, \dots, q; h_j(\mathbf{x}) = 0, j=1, 2, \dots, p; \\ & \mathbf{x} = [x_1, x_2, \dots, x_n] \in \mathbf{X} \subset \mathbf{R}^n, \mathbf{y} = [y_1, y_2, \dots, y_m] \in \mathbf{Y} \subset \mathbf{R}^m \end{aligned} \quad (1.2.5)$$

式中， $\mathbf{x} = [x_1, x_2, \dots, x_n] \in \mathbf{X} \subset \mathbf{R}^n$ 称为决策向量， \mathbf{X} 是 n 维的决策空间； $\mathbf{y} = [y_1, y_2, \dots, y_m] \in \mathbf{Y} \subset \mathbf{R}^m$ 称为目标函数， \mathbf{Y} 是 m 维的目标空间；目标函数 \mathbf{F} 定义了映射函数和同时需要优化的 m 个目标； $g_i(\mathbf{x}) \leq 0, i=1, 2, \dots, q$ 定义了 q 个不等式约束； $h_j(\mathbf{x}) = 0, j=1, 2, \dots, p$ 定义了 p 个等式约束。在此基础上，给出以下几个重要定义。

定义 1.1 可行解

对于 $\mathbf{x} \in \mathbf{X}$ ，如果 \mathbf{x} 满足约束条件 $g_i(\mathbf{x}) \leq 0, i=1, 2, \dots, q$ 和 $h_j(\mathbf{x}) = 0, j=1, 2, \dots, p$ ，则 \mathbf{x} 称为可行解。

定义 1.2 可行解集合

由 \mathbf{X} 中所有的可行解组成的集合称为可行解集合，记为 $\mathbf{X}_f (\mathbf{X}_f \subseteq \mathbf{X})$ 。

定义 1.3 Pareto-占优

对于给定的两点 $\mathbf{x}, \mathbf{x}^* \in \mathbf{X}_f$ ， \mathbf{x}^* 是 Pareto-占优（非支配）的，当且仅当式 (1.2.6) 成立，记为 $\mathbf{x}^* \succ \mathbf{x}$ 。

$$(\forall i \in \{1, 2, \dots, m\} : f_i(\mathbf{x}^*) \leq f_i(\mathbf{x})) \wedge (\exists k \in \{1, 2, \dots, m\} : f_k(\mathbf{x}^*) < f_k(\mathbf{x})) \quad (1.2.6)$$

定义 1.4 Pareto-最优解

一个解 $\mathbf{x}^* \in X_f$ 被称为 Pareto-最优解（最优解），当且仅当满足如下条件：

$$\neg \exists \mathbf{x} \in X_f : \mathbf{x} \succ \mathbf{x}^* \quad (1.2.7)$$

Pareto-最优解也称为非劣解或者有效解。所有的最优解组成的矢量集称为非支配集。这些解除了它们都在 Pareto-最优解里，没有明显联系。

定义 1.5 Pareto-最优解集

所有 Pareto-最优解组成的集合 P_s 称为 Pareto-最优解集，定义如下：

$$P_s = \{\mathbf{x}^* \mid \neg \exists \mathbf{x} \in X_f : \mathbf{x} \succ \mathbf{x}^*\} \quad (1.2.8)$$

定义 1.6 Pareto-前端

Pareto-最优解集合 P_s 中的解对应的目标函数值组成的集合 P_F 称为 Pareto-前端，定义如下：

$$P_F = \{F(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})] \mid \mathbf{x} \in P_s\} \quad (1.2.9)$$

与单目标优化相比，多目标的复杂程度大大增加了，它需要同时优化多个目标。这些目标往往是不可比较的，甚至是相互冲突的，一个目标的改善有可能引起另一个目标性能的降低。与单目标优化问题的本质区别在于，多目标优化问题的解不是唯一的，而是存在一个最优解集合，集合中的元素称为 Pareto-最优解或非支配集。Pareto-最优解就是指不存在比其中至少一个目标好而其他目标不劣的更好的解，也就是说，不可能通过优化其中部分目标而其他目标不至劣化。Pareto-最优解集中的元素就所有目标而言是彼此不可比较的。由于多目标优化问题一般不存在单个最优解，因此希望求出其 Pareto-最优解集，根据 Pareto-前端的分布情况进行多目标决策^[30]。

自 1950 年以来，用于解决多目标优化问题的方法层出不穷，分为传统方法和智能计算算法两大类。其中如加权和法、 ε -约束法、参考目标距离法、Benson 法等传统方法在解决多目标优化问题时，存在着很多缺点，如目标之间性质不同，单位不统一，不易作比较；加权值的选取带有较大主观性；目标之间相互

矛盾, 加权目标函数拓扑结构复杂等。而智能算法如遗传算法、蜂群算法、粒子群优化算法等不要求目标函数连续、可微等信息, 具有较好的全局寻优能力而受到研究人员的关注。

1.2.4 多目标优化方法简介

传统的多目标优化方法是多目标问题的各个子目标聚合成一个带正系数的单目标函数, 系统由决策者决定或者由优化方法自适应调整。为获得近似的 Pareto-最优解集, 一些优化方法是用不同的系数来实现动态优化。常见的古典方法有线性加权和法、约束法、目标规划法、极大极小法等, 以下简单介绍前两种方法。

1. 线性加权和法

该方法为每个目标函数分配权重并将目标函数和权重组合为单一目标。加权和方法可表示为

$$\min f(\mathbf{x}) = \sum_{i=1}^m \omega_i f_i(\mathbf{x}), s.t.: \mathbf{x} \in X_f \quad (1.2.10)$$

式中, ω_i 为权值, 不失一般性, 通常可以将权值正则化后使得 $\sum_{i=1}^m \omega_i = 1$ 。该方法的优点是容易理解, 便于计算; 缺点是权重无法确定。此外, 加权和法对 Pareto-前端的形状很敏感, 不能处理其前端的凹部, 而且多目标之间往往不可比较, 从而限制其应用。加权和方法的关键是如何分配权重, 许多学者最近提出了若干权重调整的方法, 如固定权重方法、随机权重方法和适应性权重方法等。

在固定权重方法中, 权重在整个进化过程中不会改变, 权重可由事先获得的有关目标知识确定。在随机权重方法中, 选择过程中的每一步都随机给出权重, 这样可以给所有可能的组合以平等的机会。虽然在统计上这种方法对 Pareto 边界给出了均匀的选择压力, 但它忽略了每个 Pareto 解中可能获得的信息。在适应性权重方法中, 权重根据当前种群的适应性调整以获得朝向正理想点的搜索压力。由于将当前种群中的有用信息用于调整下一代的权重, 因此这种方法的选择压力介于固定权重和随机权重方法之间。

2. 约束法

该方法是一种不局限于优化最优 Pareto-前端为凸的方法。其主要思想是随机将 m 个目标中的某个目标函数作为单目标优化问题的目标函数，其余 $m-1$ 个函数作为约束条件。经约束转化后的单目标优化问题为

$$\min f(\mathbf{x}) = f_j(\mathbf{x}), \text{s.t.} : e_i(\mathbf{x}) = f_i(\mathbf{x}) - \varepsilon_i, 1 \leq i \leq m, i \neq j, \mathbf{x} \in X_f \quad (1.2.11)$$

该方法的优点是简单易行， ε_i 在优化工程中取不同的值，便可找到多个 Pareto-最优解，因而在实际问题中很实用；缺点是选取合适的容许值 ε_i 往往需要先验知识，但这些先验知识往往是未知的。另一方面，上述传统方法为了获得原问题的多个 Pareto-最优解往往需要多次运行求解单目标优化问题的算法，由于这些运行求解过程相互独立，它们之间的信息无法共享，导致计算的开销较大，且得到的解有可能不可比较，令决策者难以有效决策，同时造成时间及资源的浪费。因此对较复杂、高维多约束的多目标函数的多目标优化问题，应用这些传统方法往往是不可行的。

多目标优化问题不存在唯一的全局最优解，过多的非劣解是无法直接应用的，所以在求解时就是要寻找一个最终解。求最终解主要有三类方法：

(1) 生成法，即先求出大量的非劣解，构成非劣解的一个子集，然后按照决策者的意图找出最终解。

(2) 交互法，先不求出很多的非劣解，而是通过分析 with 决策者对话的方式逐步求出最终解。

(3) 权重法，即事先要求决策者提供目标之间的相对重要程度，算法以此为依据，将多目标问题转换为单目标问题进行求解。

而这些主要是通过算法来实现的，一直以来很多专家学者采用不同算法解决多目标优化问题，如多目标进化算法、多目标粒子群算法和蚁群算法、模拟退火算法及人工免疫系统。

1.3 智能计算在通信技术中的应用

很多通信技术问题都可以建模为典型的单目标或多目标优化问题，因此，将解决优化问题的智能计算方法应用于通信技术领域是可行的。研究和开发具有较高的优化性能、时间性能和初值鲁棒性的智能优化算法并将其应用于通信技术难题中，对降低通信系统的建设成本及推进未来移动通信系统的发展均具有积极的意义。对通信系统和智能计算算法的实时性和有效性的追求是一个不断追求完美的过程，尽管现在国内外对智能计算及其在通信技术中应用的研究论文还在不断出现，但这个领域的研究空间还很大，人们不断地把新的计算方法和技术应用到通信技术中，为该研究方向的理论研究和实用化发展提供新思路和新方法。

1.3.1 简介

1. 多用户检测

CDMA 通信系统是一种严重受干扰限制的系统，多址干扰和远近效应是这种通信系统很难避免的主要干扰。如何有效地抑制多址干扰和远近效应，提高通信系统性能和容量，具有很重要的理论价值和现实意义。解决这些问题的一种有效方法就是在接收端使用多用户检测技术。多用户检测不是把多址干扰和远近效应简单地看作干扰噪声来处理，而是把它们作为一种有用的信息，充分地利用各用户间的关联进行联合检测，提高系统的检测性能和系统容量，因此多用户检测成为 CDMA 通信系统的一个关键技术。

为了使多用户检测技术实用化，各国的研究者提出许多计算复杂度相对较低的次优多用户检测器。次优多用户检测器一般可分为线性检测器和非线性检测器两大类。经典的线性检测器主要包括去相关检测器和线性最小均方差检测器，尽管实现简单，但抗多址干扰和远近效应能力不强。而非线性检测器主要

包括干扰消除检测器^[31]、基于神经网络的准最优多用户检测器^[32]和基于智能优化算法的准最优多用户检测器^[33-35]。

随着智能计算理论的兴起,在各种扩频通信系统设计智能多用户检测器引起人们的广泛关注^[36-38]。这是因为基于智能信息处理的多用户检测能在各种系统和各种噪声环境下达到近似最优的性能,并且空时处理的多用户检测和空时编码的多用户检测也可以使用智能计算算法来实现。因此,只要一种新的智能计算方法问世,就会有研究者将其应用到多用户检测器的设计上。

2. 决策引擎和系统参数调整

认知无线电将人工智能技术与无线电技术相结合,其智能性主要依赖于认知决策引擎这一关键技术。认知决策引擎相当于 CR 的“大脑”,也是 CR 的核心技术之一,所以首要的研究是如何完成决策过程。第一个需要探讨的问题是认知决策过程应该以分布式方式还是集中式方式来完成^[39]。智能性通常是基于分布式系统实现的,这对于实现认知网络是至关重要的,且认知无线电的决策过程是由不同设备的协作完成的,因此分布式系统成为首要选择。认知决策引擎通过从经验和环境中学习知识进行分析,然后智能地调整工作参数使其满足用户的通信需求,因此认知决策引擎可以被看作一个多目标优化问题。那么随之而来的第二个问题是如何选择决策算法。现有的决策算法有神经网络、遗传算法和蚁群算法等,选择一个符合并满足认知无线电系统要求的算法是十分重要的。

维吉尼亚工学院的无线通信中心(VC-CWT)于 2004 年设计的认知决策引擎模型被广泛接受,它是基于测试平台的 CR 仿生决策模型,利用 GA 来实现系统参数的优化^[40]。Newman 等人提出了多载波系统中的基于 GA 的认知决策引擎^[41],讨论分析了增加变量参数对实现决策引擎技术复杂度的影响^[42]。随后,又在此基础上,将前一代种群所得到的认知环感知结果作为当前代的初始信息,以加速算法收敛^[43]。与此同时,国内一些学者也将各种智能算法用于解决认知引擎决策问题中,设计了基于智能计算算法的认知决策引擎^[44-47]。上述研究均考虑的是对加权后的单目标问题进行优化,并不能有效考虑到各个通信指标的

要求。文献[48,49]将多目标算法应用于认知决策引擎,但是所提算法较为复杂,实时性不高,并不能普遍应用于认知决策引擎中。

3. 频谱分配

频谱分配主要解决的问题是如何公平而有效地为用户分配有限的空闲频谱资源以实现频谱的高效利用。根据认知用户对通信性能和服务质量等的需求,将有限的频谱资源进行合理分配,使认知无线电的整体系统性能得到改进或接近理想状态,这也是实现认知无线电的关键。因此,国内外各研究机构及学者对其投入较大的关注并进行了相关的研究。

早期关于频谱分配利用的研究包括频谱共享池^[50]、CORVUS 系统^[51]及 OCRA 网络^[52],但这些研究都是假设在一定理想的条件下,许多因素对系统的影响并没有考虑,近年来许多考虑实际通信环境的频谱分配算法被提出。其主流频谱算法根据建立模型的差异可概括为以下几类:基于图论的方法^[53]、基于拍卖竞价分配方法^[54]、博弈论^[55]、干扰温度模型^[56]和竞争定价模型^[57]。

在上述各种方法中,基于图论的方法已被广泛应用于各工程领域,在解决频谱资源分配的问题上也获得了较好的结果。Zheng 等人提出的 CSGC 算法考虑了实际通信中的可用信道和干扰限制,并利用传统图着色中冲突图的顶点模拟实际的认知用户,进而构造出用于解决频谱分配的数学模型。但传统的 CSGC 方法存在运算量过大等缺点。为此,许多学者在 CSGC 模型的基础上继续深入研究。文献[58]将并行算法引入到 CSGC 方法中,在获得与 CSGC 相近性能的同时降低了运算量和时间开销,并能适应快变的环境。

随着智能计算的兴起,人们发现可以将智能算法应用于解决基于着色论模型频谱分配问题。学者们纷纷将遗传算法^[59]、量子遗传算法^[60]、粒子群算法^[61]、混合蛙跳算法^[62]、免疫克隆优化^[63]和蚁群优化^[64]等应用到解决频谱分配问题中,与 CSGC 算法相比能够获得更优秀的性能,实现网络效益最大化。

4. 频谱感知

频谱感知作为认知无线电的一项关键技术,它的主要功能在于检测可供认

知用户使用的频谱空穴，同时监测主用户信号活动情况，保证主用户再次使用频谱时，认知用户能快速退出相应频段。目前已提出的频谱感知方法主要包括匹配滤波器检测、能量检测、周期平稳特征检测及多分辨率频谱感知^[65]，这些频谱感知方法均为单节点感知方法。然而，在阴影和深度衰落情况下，单个节点的感知结果并不可靠。因此，需要对多个节点的感知结果进行融合，以提高其检测可靠性，即协作频谱感知技术^[66]。文献[67]提出了一种线性协作感知框架，但如何在最短的时间内求解出最优权向量是一个优化难题。为了解决该难题，智能优化等算法都可被用于去求解频谱感知的最优权向量或频谱感知决策^[68-73]。

5. 测向技术

测向又称为空间谱估计或者波达方向（Direction of Arrival, DOA）估计，是阵列信号处理的一个重要研究方向，在信息战、雷达、被动声纳、导航、地震学及移动和卫星系统等军事和民用领域都有着广泛的应用，具有广阔的应用前景，极大地吸引了国内外学者的研究目光。DOA 估计技术以空域处理和时域处理为基础，经过近半个世纪的研究和发展，提出了一系列性能优良的超分辨测向算法^[74]。经典的测向算法主要包括子空间分解类算法^[75,76]、极大似然算法^[77,78]和子空间拟合算法^[79]。子空间分解类算法计算量少，但需要预处理才可测相干信源；而极大似然算法和子空间拟合算法计算量大，但具有天然的解相干能力和高精度。随着测向技术的发展，对测向算法的工程应用和环境提出了新的要求。因此，为了满足不同测向环境的需求，分别针对相干信号、低计算复杂度、任意阵列结构、高阵列利用率及冲击噪声背景等具体条件提出了相应的改进算法^[80-83]。

随着现代信号处理技术的发展，现代信号处理在空间谱估计中的应用已成为空间谱估计技术新的发展方向。由于神经网络在降低计算量方面有优势，便产生了一系列基于神经网络的 DOA 估计方法，如自组织模糊神经网络^[84]、径向基神经网络及基于高维仿生信息学的神经网络^[85]等。神经网络方法在空间谱估计中的应用避免了数据矩阵的特征分解，但神经网络算法的缺点是网络结构比较复杂、算法需要较长的时间学习等，并且一些好的网络结构需要充足的样

本训练才能获得,测向装置需事先存储训练好的各种情况和环境下的神经网络结构,因此在实际应用中受限较多。在空间谱估计中,特别是多维最优值搜索花费了大量的时间,而遗传算法^[86]等智能优化算法在最优值搜索过程中会减少算法的计算量,在解决一些简单的测向目标函数时显示了较好的性能,但在解决复杂测向目标函数时容易早熟收敛,且这类算法还不够完善,有待于进一步研究^[87-90]。

1.3.2 展望

1. 多用户检测

早期的多用户检测技术是在高斯白噪声环境下进行的研究,然而,随着实用化多用户检测技术的逐渐受重视,可以看到研究实际多址信道下的多用户检测技术有着更重要的实际意义。如在非高斯信道和冲击噪声信道等更恶劣的噪声环境下,一些高斯环境下经典的多用户检测器性能严重恶化,经典的最优多用户检测器已经不再是最优的,需要针对这些信道设计鲁棒多用户检测器。

国内外的研究表明,当前多径衰落 MC-CDMA 等系统并没有从根本上解决多址干扰、远近效应问题。所以,研究全新的盲自适应的智能 MC-CDMA 多用户检测技术,并进一步地把新提出的信号处理算法移植到其他无线通信系统,例如 UWB 系统和认知无线电系统,对这些无线通信系统的发展也会有促进性的影响。若能进一步地把智能天线技术和认知无线电理论引入多用户检测中,在冲击噪声环境下设计基于智能计算方法的认知鲁棒多用户检测技术,将成为一个富有前景的研究方向。

2. 决策引擎和系统参数调整

现有的认知决策引擎的设计大多仅考虑了 3 个参数的调整过程,实际通信环境对很多通信指标都有要求,因此可以建立一个包含更多目标的优化问题模型。并且实际的通信环境是复杂多变的,参数调整过程也应是实时变化的,故可对动态引擎决策过程进行进一步研究。

将绿色认知无线电参数调整这一有约束优化的决策引擎问题转化为无约束单目标优化问题可进行高精度求解,把约束条件以惩罚因子的形式体现在优化的问题模型中,这不能充分地考虑约束条件的限制,因此,可以设计一个适用于有约束条件的多目标函数和多目标优化算法来进行绿色认知无线电的参数调整。

3. 频谱分配

静态频谱分配被广泛研究,它假设在频谱分配的过程中认知用户和授权用户的位置是不发生变化的,而实际上可能存在认知用户和授权用户位置随时间变化的情况,因此需要在动态频谱分配过程中进行更深入的研究。另外,绿色认知无线电系统的能量高效利用和频谱分配的综合考虑还需进行深入研究,在实现绿色通信技术时,频谱资源的高效利用也是极为重要的一个方面。能量的高效利用与频谱资源的高效利用在很多时候是冲突的,如何在二者之间做出权衡需要进行进一步研究。

4. 频谱感知

协作感知的性能与协作用户数量、各用户门限值的确定及位置分布情况等因素密切相关。如何选取这些协作感知参数以获得最佳的检测性能也是协作感知研究的重要内容。

对于现阶段的频谱感知模型,无论是单节点感知还是协作感知,都采用高斯噪声模型,但是对于突发噪声的特殊情况欠缺研究。因此,对于冲击噪声背景下的频谱感知方法的研究也将是频谱感知的一项重要内容。

现在的研究大多集中在单个感知用户网络参与协作的情形,基于网络层的多感知用户网络间的协作也可能是未来研究的一个方向。

5. 测向技术

在测向技术的发展过程中,信源数与结构估计所面临的突出问题是对信源个数和结构估计算法的落后已阻碍了测向方法的实用化进程,在低信噪比、少

阵元数（阵元数小于信源数）和非直线阵情况下，研究信源结构和个数估计的有效算法是富有挑战的开放性难题。

如何找到适用于各种阵列结构并能实现阵列扩展（信源数大于阵元数），计算量适中，还可同时估计相干信源和非相干信源，并且性能优于加权子空间拟合测向算法的测向方法是值得努力的方向。在任意阵测向中突破空间谱估计算法的相干和独立信源的自由度限制，必将促进该领域的进一步发展。

因此，研究强冲击噪声背景下的测向问题，提出可应用于高斯噪声、弱冲击噪声和强冲击噪声的更广义的测向理论，是非常有价值的研究方向。设计新的量子群智能方法解决现有通信技术中所遇到的难题，打破其应用局限，必将促进通信技术和量子群智能进一步的发展。随着智能计算和量子群智能的不断发展和深入，其理论必将越来越成熟，内容也会越来越丰富，应用前景也会越来越广阔。

1.4 本书内容和结构安排

第 1 章简单介绍量子群智能及其在通信技术中的应用，使读者了解本书的整体概貌。

第 2 章介绍了解决离散优化问题的量子群智能算法，介绍了 3 种离散量子群智能算法，并与经典离散智能计算算法进行了性能测试和比较。所介绍的 3 种离散优化算法是：离散量子粒子群算法、离散量子蜂群算法和离散量子细菌觅食算法。

第 3 章介绍了解决连续优化问题的量子群智能算法，介绍了 3 种连续量子群智能算法，并与经典连续智能计算算法进行了性能测试和比较。所介绍的 3 种连续优化算法分别是：连续量子蛙跳算法、连续量子文化蛙跳算法和量子细菌觅食算法。

第 4 章首先介绍了 CDMA 多用户系统接收信号的数学模型和典型的多用户

检测方法。然后结合量子优化算法和智能计算,分别设计了3种基于量子智能计算的多用户检测方法:免疫克隆量子算法的多用户检测、量子神经网络多用户检测和基于量子蜂群算法的鲁棒多用户检测。

第5章首先基于量子计算和膜计算理论,主要介绍了一种膜量子蜂群算法,并将其应用于认知决策引擎技术中,有针对性地解决了现有认知决策引擎方法所存在的一些的局限和难题,并提高了整个系统的性能。然后介绍了绿色认知无线电的参数调整模型,它是认知决策引擎的进一步延伸。进而提出了一种基于量子细菌觅食算法的参数调整方法,仿真实验证明了所提方法能够在保证用户服务质量的前提下实现绿色通信技术。

第6章首先将量子计算引入粒子群算法,设计了基于量子粒子群算法的认知无线电频谱分配,其相对于经典的算法具有更高的网络效益。其次,详细介绍了多目标膜量子蜂群算法,包括非支配解排序操作和拥挤度计算操作。再次,建立了多目标频谱分配模型,并根据此模型提出了基于多目标膜量子蜂群算法的频谱分配方法。最后通过仿真实验验证了所提频谱分配方法能够有效地兼顾用户所获得的效益与公平性。

第7章介绍了如何将所设计的量子细菌觅食算法应用于解决认知协作感知问题,并进行了计算机仿真。仿真结果表明,基于量子细菌觅食算法的频谱感知方法相对于基于经典智能算法的频谱感知方法具有很大优势。

第8章在高斯噪声环境下介绍了一些测向方法,针对两个高斯噪声环境下的测向难题,在不同的应用背景下提出了两种智能优化测向方法:基于文化量子算法的广义高阶累计量测向算法和基于量子文化蛙跳算法的非圆信号测向算法,解决了高阶累计量测向和非圆测向中遇到的应用局限,可有效实施测向。基于文化量子算法的广义加权子空间拟合测向使基于四阶累积量的测向技术在不损失阵列孔径的情形下实现了对相干信源测向。基于量子文化蛙跳算法的非圆极大似然测向算法更好地利用信号的非圆信息,可测相干信源和扩展阵列孔径。

第9章首先介绍了3种低阶矩阵,然后在冲击噪声环境下介绍了基于量子文化细菌觅食优化算法的非圆信号的极大似然测向方法,这种方法可以高精度

求解非圆信号的相干和非相干信号源测向问题。最后针对冲击噪声背景下的动态目标测向问题，连续量子粒子群算法和一种极大似然算法被设计和应用到动态目标测向问题中。

参考文献

- [1] 高洪元. 多用户检测中智能信息处理理论研究. 哈尔滨工程大学博士学位论文.
- [2] Marwala T. Condition monitoring using computational intelligence methods: applications in mechanical and electrical systems. Springer, 2012.
- [3] 崔志华, 曾建潮. 微粒群优化算法. 北京: 科学出版社, 2011.4.
- [4] Diwold K, Aderhold A, Scheidler A, et al. Performance evaluation of artificial bee colony optimization and new selection schemes. Memetic Computing, 2011, 3(3): 149-162.
- [5] Das S, Dasgupta S, et al. On stability of the chemotactic dynamics in bacterial foraging optimization algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2009, 39(3): 670-679.
- [6] Eusuff M M, Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.
- [7] M Ohya, N Masuda. NP problem in quantum algorithm. Los Alamos Physics preprint archive, <http://xxx.lanl.gov/lanl-e-print-quant-ph/9809075>, 1998.
- [8] Deutsch D. Quantum computational networks. Proceedings of the Royal Society, London A, 1989, 425: 73-90.
- [9] Deutsch D, Jozsa R. Rapid solution of the problems by quantum computation.

- Proceedings: Mathematical and Physical Sciences, 1992, 439(1907): 553-558.
- [10] 李承祖. 量子通信和量子计算. 长沙: 国防科学大学出版社, 2000: 148-184.
- [11] Narayanan A. Quantum computing for beginners. Proceedings of the 1999 Congress on Evolutionary Computation, 1999: 2231-2238.
- [12] Spector L, Barnum H, Bernstein H J, Swamy N. Finding a better-than-classical quantum AND/OR algorithm using genetic programming. Proceedings of the 1999 Congress on Evolutionary Computation, 1999: 2239-2246.
- [13] Moore M P. Quantum-inspired algorithms and a method for their construction. Master's thesis, Department of Computer Science, University of Exeter, Exeter, UK, 1995.
- [14] Han K H, Kim J H. Genetic Quantum algorithm and its application to combinatorial optimization problem. Proceedings of the 2000 IEEE Congress on Evolutionary Computation, 2000: 1354-1360 .
- [15] Han K H, Kim J H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Transactions on Evolutionary Computation, 2002, 6(6):580-593.
- [16] Yang J A, Li B, Zhuang Z Q. Multi-universe parallel quantum genetic algorithm its application to blind-source separation. Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, 2003, 1: 393 -398.
- [17] 杨淑媛, 刘芳, 焦李成. 量子进化策略. 电子学报, 2001, 29 (12A): 1873-1877.
- [18] 张葛祥, 李娜, 金炜东. 一种新量子遗传算法及其应用. 电子学报, 2004, 32 (3): 476-479.
- [19] 李阳阳, 焦李成. 求解 SAT 问题的量子免疫克隆算法. 计算机学报, 2007,

30 (2): 176-183.

- [20] Yang S Y, Wang M, Jiao L C. A genetic algorithm based on quantum chromosome. 2004 7th International Conference on Signal Processing, 2004, 2:1622-1625.
- [21] Guo R H, Li B, Zhou Y, Zhuang Z Q. Hybrid quantum probabilistic coding genetic algorithm for large scale hardware-software co-synthesis of embedded systems. IEEE Congress on Evolutionary Computation, 2007: 3454-3458P.
- [22] De Oliveira L D, Ciriaco F, Abrao T, Jeszensky P J E. Particle swarm and quantum particle swarm optimization applied to DS/CDMA multiuser detection in flat rayleigh channels. 2006 IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications, 2006:133 -137.
- [23] Gao Hongyuan, Cui Wen, Li Chenwan. A quantum bacterial foraging optimisation algorithm and its application in spectrum sensing. International Journal of Modelling, Identification and Control. 2013, 18(3): 234-242.
- [24] 李士勇, 李盼池. 量子计算与量子优化算法. 哈尔滨: 哈尔滨工业大学出版社, 2009.
- [25] Chung C Y, Yu H, Wong K P. An advanced quantum-inspired evolutionary algorithm for unit commitment. IEEE Transactions on Power Systems, 2011, 26(2): 847-854.
- [26] 孙俊, 方伟, 吴小俊, 等. 量子行为粒子群优化: 原理及其应用. 北京: 清华大学出版社, 2011.
- [27] 李晨琬. 基于量子群智能的认知无线电关键技术研究. 哈尔滨工程大学硕士学位论文, 2015.
- [28] 魏静萱. 解决单目标和多目标优化问题的进化算法. 西安电子科技大学博士学位论文, 2009.

- [29] 雷德明, 严新平. 多目标智能优化算法及其应用. 北京: 科学出版社, 2009.3.
- [30] 焦李成, 尚荣华, 马文萍, 等. 多目标优化免疫算法、理论和应用. 北京: 科学出版社, 2010.1.
- [31] Kohno R, et al. Combination of an adaptive array antenna and a canceller of interference for direct-sequence spread-spectrum multiple-access systems, IEEE JASC, 1990, 18(4): 675-682.
- [32] Kechriotis G, Manolakos E S. Hopfield neural network implementation in the optimal CDMA multiuser detector. IEEE Trans. Neural Networks, 1996, 7(1): 131-141.
- [33] Ergun C, Hacioglu K. Multiuser detection using a genetic algorithm in CDMA communications systems. IEEE Trans. Commun, 2000, 48(8): 1374-1383.
- [34] Wen J H, Chang C W, Hung H L. Blind multiuser detection in frequency domain for MC-CDMA systems using particle swarm optimization. Wireless Personal Communications. 2010, 54 (3): 447-466.
- [35] 刘洪武. 空时分组码系统的优化传输与多用户检测技术研究. 西南交通大学博士论文, 2008.
- [36] Gong Maoguo, Jiao Licheng, Ma Wenping & MA JingJing. Intelligent multi-user detection using an artificial immune system. Science in China, 2009, 52(12): 2342-2353.
- [37] 刘洪武, 冯全源. 分集接收的 STBC-MC-CDMA 系统中基于 PSO 算法的多用户检测. 电子与信息学报, 2009, 31 (1): 45-48.
- [38] Hongwu Liu, Quanyuan Feng. Particle swarm optimization-based and receive-diversity-aided multiuser detection for STBC MC-CDMA systems. IEEE on Signal Processing Letters, 2009, 39(1): 45-48.

- [39] Villain F, Le Guillou Y. Cognitive radio systems: Overview and challenges. 3rd IEEE International Conference on Awareness Science and Technology (iCAST), 2011: 497-502.
- [40] Rieser C J, Rondeau T W, Bostian C W, et al. Cognitive radio testbed: further details and testing of a distributed genetic algorithm based cognitive engine for programmable radios. IEEE Military Communications Conference, 2004: 1437-1443.
- [41] Newman T R, Barker B A, Wyglinski A M, et al. Cognitive engine implementation for wireless multicarrier transceivers. Wireless communications and mobile computing, 2007, 7(9): 1129-1142.
- [42] Newman T R, Evans J B. Parameter sensitivity in cognitive radio adaptation engines. 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2008, 756-760.
- [43] Newman T R, Rajbanshi R, et al. Population adaptation for genetic algorithm-based cognitive radios. Mobile Networks and Applications, 2008, 13(5): 442-451.
- [44] 赵知劲, 郑仕链, 尚俊娜, 等. 基于量子遗传算法的认知无线电决策引擎研究. 物理学报, 2007, 56 (11): 6760-6766.
- [45] 郑仕链, 赵知劲. 基于模拟退火遗传算法的认知无线电决策引擎研究. 计算机仿真, 2005, 1 (25): 192-196.
- [46] 赵知劲, 徐世宇, 郑仕链, 等. 基于二进制粒子群算法的认知无线电决策引擎. 物理学报, 2009, 58 (7): 5118-5125.
- [47] Zhang J, Zhou Z, Gao W, et al. Cognitive radio adaptation decision engine based on binary quantum-behaved particle swarm optimization. 6th IEEE International ICST Conference on Communications and Networking in China (CHINACOM), 2011: 221-225.

- [48] Guo Q W, Jin L L, Min Z, et al. Solving performance optimization problem of cognitive radio with multiobjective evolutionary algorithm. *Computer Engineering and Applications*, 2007, 43(20): 159-162.
- [49] Yong L, Hong J, Qing H Y. Design of cognitive radio wireless parameters based on multi-objective immune genetic algorithm. *IEEE International Conference on Communications and Mobile Computing*, 2009, 1: 92-96.
- [50] Tang H Y. Some physical layer issues of wide-band cognitive radio system. *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*. Baltimore, 2005: 151-159.
- [51] Robert W B, Adam W, Danijela C, Shridhar M M, Daniel W. CORVUS: a cognitive radio approach for usage of virtual unlicensed spectrum. UC Berkeley: Berkeley Wireless Research Center, 2004.
- [52] Akyildiz I F, Lee W, Vuran M, et al. Next generation/dynamic spectrum access cognitive radio wireless networks: a survey. *Computer Networks*, 2006, 50(13): 2127-2159.
- [53] Peng C, Zheng H, Zhao B Y. Utilization and fairness in spectrum assignment for opportunistic spectrum access. *Mobile and Networking Technologies for Social Applications*, 2006, 11(4): 555-576.
- [54] Gandhi S, Buragohain C, Cao L, et al. A general framework for wireless spectrum auctions. *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, 2007: 22-33.
- [55] Etkin R, Parekh A, Tse D. Spectrum sharing for unlicensed bands. *IEEE New Frontiers in Dynamic Spectrum Access Networks*. Baltimore, 2005, 25(3): 517-528.
- [56] Clancy T, Arbaugh W. Measuring interference temperature. *Virginia Tech Symposium on Wireless Personal Communications*, 2006: 1-10.

- [57] Nivato D, Hossain E. Competitive pricing for spectrum sharing in cognitive radio networks: Dynamic game, inefficiency of Nash equilibrium, and collusion. *IEEE Journal on Selected Areas in Communications*, 2008, 26 (1): 192-202.
- [58] 廖楚林, 陈劼, 唐友喜, 等. 认知无线电中的并行频谱分配算法. *电子与信息学报*, 2007, 29 (7): 1608-1611.
- [59] Zhao Z, Peng Z, Zheng S, et al. Cognitive radio spectrum allocation using evolutionary algorithms. *IEEE Transactions on Wireless Communications*, 2009, 8(9): 4421-4425.
- [60] 赵知劲, 彭振, 郑仕链, 等. 基于量子遗传算法的认知无线电频谱分配. *物理学报*, 2009, 58 (2): 1358-1363.
- [61] Zhang B W, Hu K Y, Zhu Y L. Spectrum allocation in cognitive radio networks using swarm intelligence. *2nd International Conference on Communication Software and Networks*. 2010: 8-12.
- [62] 彭振, 赵知劲, 郑仕链. 基于混合蛙跳算法的认知无线电频谱分配. *计算机工程*, 2010, 36 (6): 210-217.
- [63] 柴争义, 刘芳. 基于免疫克隆选择优化的认知无线网络频谱分配. *通信学报*, 2010, 31 (11): 92-100.
- [64] 杨淼, 安建平. 认知无线网络中一种基于蚁群优化的频谱分配算法. *电子与信息学报*, 2011, 33 (10): 2306-2311.
- [65] GU Jun-rong, Jang S J, Kim J M. A proactive dynamic spectrum access method against both erroneous spectrum sensing and asynchronous inter-channel spectrum sensing. *KSII Transactions on Internet and Information Systems*. 2012, 6(1): 361-378.
- [66] Cabric D, Mishra S M, Brodersen R. Implementation issues in spectrum sensing for cognitive radios. In *Proc. 38th Asilomar Conf. Signals, Systems*

- and Computers, Pacific Grove, CA, United states: Institute of Electrical and Electronics Engineers Computer Society, 2004: 772–776.
- [67] QUAN Zhi, CUI Shu-quan, Sayed A H. Optimal linear cooperation for spectrum sensing in cognitive radio networks. *IEEE Journal of Selected Topics in Signal Processing*, 2008, 2(1): 28-40.
- [68] 刁鸣, 钱荣鑫, 高洪元. 狼群优化的神经网络频谱感知算法. *计算机工程与应用*, 2015.1.
- [69] 江辉, 陈飞飞, 杜文峰. 基于改进群搜索优化算法的认知无线电协作频谱感知. *电路与系统学报*, 2013.187-192.
- [70] Pradhan P M, Panda G, Majhi B. Multiobjective cooperative spectrum sensing in cognitive radio using cat swarm optimization. *Wireless Advanced(WiAd)*, 2012, 44-48.
- [71] 郑仕链, 楼才义, 杨小牛. 基于改进混合蛙跳算法的认知无线电协作频谱感知. *物理学报*, 2010, 59 (5): 3611-3617.
- [72] 高洪元, 崔闻. 文化蛙跳算法及其在频谱感知中的应用. *中南大学学报*, 2013, 44 (9): 3723-3730.
- [73] Gao Hongyuan, Cui Wen, Jiang Yilin. A quantum-inspired shuffled frog leaping algorithm and its application in cognitive radio. *International Journal of Digital Content Technology and its Applications*, 2012, 20(6):32-42 .
- [74] 王永良. 空间谱估计理论与算法. 北京: 清华大学出版社, 2004.11.
- [75] Schmidt R O. Multiple emitter location and signal parameter estimation. *IEEE Trans*, 1986, AP-34(3): 276-280.
- [76] Roy R, Kailath T. ESPRIT-estimation of signal parameters via rotational invariance techniques. *Acoustics, Speech, and Signal Processing* , *IEEE Transactions on* Volume 37, Issue 7, July 1989 Page(s): 984-995.

- [77] Stoica P, Nehorai A. MUSIC, Maximum likelihood, and Cramer-Rao bound. In Proc. ICASSP, 1988: 2296-2299.
- [78] Ottersten B, Viberg M, Stoica, Nehorai A. Exact and large sample ML techniques for parameter estimation and detection in array processing. IN Haykin, Litva, and shepherd, editors, Radar array processing , Springer-Verlag , Berlin,1993: 99-151.
- [79] Viberg M, Ottersten B. Sensor array processing based on subspace fitting. IEEE Transactions on Signal Processing, 1991, 39 (5): 1110-1121.
- [80] 刁鸣, 高璐, 高洪元, 等. 基于非均匀线阵的压缩感知波达方向估计. 计算机工程, 2015, 41 (10): 83-87.
- [81] 刁鸣, 李晓刚, 高洪元. 基于四阶累积量的最大似然测向方法研究. 电子信息对抗技术, 2007, 22 (5): 4-6.
- [82] 何昭然, 高洪元. 文化鱼群算法的广义 MUSIC 测向技术. 解放军理工大学学报: 自然科学版, 2014, 15 (1): 7-11.
- [83] 安春莲, 赵振宇, 刁鸣, 等. 独立信号与相干信号并存的任意阵列测向新方法. 哈尔滨工程大学学报, 2013, (4): 517-523.
- [84] Shieh Ching-Sung, Lin Chin-Teng. Direction of arrival estimation based on phase differences using neural fuzzy network. Antennas and Propagation, IEEE Transactions on Volume 48, Issue 7, July 2000 Page(s): 1115 – 1124.
- [85] 安冬, 王守觉. 基于仿生模式识别和 PCA/ICA 的 DOA 估计方法. 电子学报, 2004, 32 (9): 1448-145.
- [86] Jin Yong, Huang Jianguo, Zhao Jianjun. Wideband Beam-Space DOA Estimation Based on Genetic Algorithm. Natural Computation, 2007. ICNC 2007. Third International Conference on Volume 3, 24-27 Aug. 2007 Page(s): 791-798.

- [87] 庞伟正, 高洪元, 王艳丽. 基于粒子群优化算法的相干信源波达方向估计. 哈尔滨工程大学学报, 2006, 27 (3): 453-456.
- [88] 刁鸣, 高洪元, 王艳温. 二维相干信源测向新方法研究. 系统工程与电子技术, 2006, 28 (9): 1289-1293.
- [89] Gao Hongyuan, Xu Chong Qiang, Li Chenwan. Quantum-inspired cultural bacterial foraging algorithm for direction finding of impulse noise, International Journal of Innovative Computing and Applications, 2014, 6(1): 44-54.
- [90] 高洪元, 刁鸣. 文化量子算法实现的广义加权子空间拟合测向. 电波科学学报, 2010, 25 (4): 798-80.

第 2 章

用于离散优化问题的量子群智能计算

由于工程中的很多问题可看作离散的组合优化问题，但因其易被描述却很难被解决而一直吸引着学者们的注意，其中很多问题延伸到实际应用中可称之为 NP 难题，传统的方法很难在短时间内求得最优解或者近似最优解。多年来，在解决复杂的组合优化问题时，越来越多的人将目光转向自然界以寻求灵感。

遗传类算法就是受达尔文进化理论的启发所设计的优化算法，它作为成熟的启发算法，现已得到广泛应用。但遗传类算法的收敛需要严格的条件限制，所以近年来一些研究者致力于探索新的启发算法。近年来，基于生物群体的优化算法有了新的发展，产生了模拟生物群体觅食本能行为的群集智能算法。群体智能是由简单个体组成的生物群落与环境及个体之间为了生存需要而发展出的一种互动行为，群体智能在没有集中控制并且不提供全局模型的前提下为寻找复杂的分布式问题的解决方案提供了平台。粒子群算法是继遗传算法之后，另一种被广泛应用的智能计算算法，得到各领域研究人员的深入研究和关注。

本章针对离散优化问题，分别介绍了量子粒子群、量子蜂群和量子细菌觅食算法，并使用测试函数进行测试，与一些没有使用量子编码的智能优化算法进行比较。量子粒子群算法和量子蜂群算法分别使用两种量子编码方式实现：一种是用一对实数表示的量子比特，称之为双链编码；另一种是使用 $[0,1]$ 之间

的一个正实数表示的模拟量子比特，可以在演化过程中构造出另一个位而节省存储空间，称之为单链编码。双链编码具有形象直观的优点，而单链编码具有节省存储空间和演化计算量的优点。尽管使用离散优化算法可以解决连续优化问题，在编码的时候存在量化误差，但是连续优化问题的经典测试函数还是可以用来测试离散优化算法性能的优劣。

2.1 量子粒子群算法

尽管离散粒子群算法是解决离散优化问题的经典算法，但其收敛性能仍具有较大的改进空间。本节分别介绍了双链编码和单链编码的离散量子粒子群算法，使用量子编码的粒子群算法更多的是借鉴连续粒子群优化算法的思想，并通过测试函数的仿真试验说明了量子编码的粒子群算法具有性能上的优势。

2.1.1 双链编码的量子粒子群算法

量子粒子的量子速度是由量子比特表示的^[1]。量子比特为最小信息单位，是一个复数对，可表示为 $(\alpha, \beta)^T$ ，并且满足 $|\alpha|^2 + |\beta|^2 = 1$ 。 $|\alpha|^2$ 给出了量子位被确定为“0”状态的概率， $|\beta|^2$ 给出了量子位被确定为“1”状态的概率。第 i 个量子粒子的量子速度定义为

$$\mathbf{v}_i^t = [\mathbf{v}_{i1}^t, \mathbf{v}_{i2}^t, \dots, \mathbf{v}_{iD}^t] = \begin{bmatrix} \alpha_{i1}^t, \alpha_{i2}^t, \dots, \alpha_{iD}^t \\ \beta_{i1}^t, \beta_{i2}^t, \dots, \beta_{iD}^t \end{bmatrix} \quad (2.1.1)$$

其中， $|\alpha_{id}^t|^2 + |\beta_{id}^t|^2 = 1, (d = 1, 2, \dots, D)$ ， α_{id}^t 和 β_{id}^t 分别被定义为 $0 \leq \alpha_{id}^t \leq 1$ ， $0 \leq \beta_{id}^t \leq 1$ 。

量子粒子的进化过程也就是量子速度的更新过程。第 i 个量子粒子的第 d 个量子旋转角所对应的量子旋转门定义为 $U(\theta_{id}^{t+1}) = \begin{bmatrix} \cos \theta_{id}^{t+1} & -\sin \theta_{id}^{t+1} \\ \sin \theta_{id}^{t+1} & \cos \theta_{id}^{t+1} \end{bmatrix}$ ，量子比特 \mathbf{v}_{id}^t 用量子旋转门 $U(\theta_{id}^{t+1})$ 更新，第 i 个量子粒子的第 d 个量子比特 \mathbf{v}_{id} 的量子旋

转门更新过程如下:

$$\mathbf{v}_{id}^{t+1} = \text{abs}(\mathbf{U}(\theta_{id}^{t+1})\mathbf{v}_{id}^t) = \text{abs}\left(\begin{bmatrix} \cos \theta_{id}^{t+1} & -\sin \theta_{id}^{t+1} \\ \sin \theta_{id}^{t+1} & \cos \theta_{id}^{t+1} \end{bmatrix} \mathbf{v}_{id}^t\right) \quad (2.1.2)$$

其中, $\text{abs}()$ 代表取绝对值函数, 使所有量子位在其定义区间。如果量子旋转角 $\theta_{id}^{t+1} = 0$, 量子比特 \mathbf{v}_{id}^t 用量子非门 N 以某种较小的概率进行更新, 这一过程可以描述为

$$\mathbf{v}_{id}^{t+1} = N\mathbf{v}_{id}^t = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{v}_{id}^t \quad (2.1.3)$$

在量子粒子群中, 有 P 个量子粒子在 D 维空间中搜寻最优位置。第 i 个量子粒子的位置为 $\mathbf{x}_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)$, $1 \leq i \leq P$ 。第 i 个量子粒子的量子速度为 $\mathbf{v}_i^t = [\alpha_{i1}^t, \alpha_{i2}^t, \dots, \alpha_{iD}^t]$, 并且将第 i 个量子粒子迄今为止所搜到的局部最优位置记为 $\mathbf{p}_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t)$, $1 \leq i \leq P$ 。 $\mathbf{p}_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gD}^t)$ 为整个量子粒子群迄今为止所搜到的全局最优位置。

第 i 个量子粒子的量子速度的更新过程可以表示如下:

$$\theta_{id}^{t+1} = e_1(p_{id}^t - x_{id}^t) + e_2(p_{gd}^t - x_{id}^t) \quad (2.1.4)$$

$$\mathbf{v}_{id}^{t+1} = \begin{cases} N\mathbf{v}_{id}^t, & \text{若 } (p_{id}^t = x_{id}^t = p_{gd}^t \text{ 且 } r < c_1) \\ \text{abs}(\mathbf{U}(\theta_{id}^{t+1})\mathbf{v}_{id}^t), & \text{其他} \end{cases} \quad (2.1.5)$$

其中, $1 \leq d \leq D$, r 为均匀分布在 $[0,1]$ 之间的随机数, c_1 为变异概率, e_1 和 e_2 为两个常数, 分别决定了量子粒子的局部最优位置和整个量子粒子群的全局最优位置对该量子粒子的量子速度和位置的更新过程中的影响程度。第 i 个量子粒子的位置通过对量子速度的测量得到, 测量方程为

$$x_{id}^{t+1} = \begin{cases} 1, & r_{id}^{t+1} > (\alpha_{id}^{t+1})^2 \\ 0, & r_{id}^{t+1} \leq (\alpha_{id}^{t+1})^2 \end{cases} \quad (2.1.6)$$

其中, r_{id}^{t+1} 为均匀分布在 $[0,1]$ 之间的随机数, $1 \leq d \leq D$, $(\alpha_{id}^{t+1})^2$ 描述了量子位 \mathbf{v}_{id}^{t+1} 出现 “0” 状态的概率。

双链编码的量子粒子群算法的步骤如下。

Step 01 初始化量子粒子群, 将量子粒子的量子速度 \mathbf{v}_i^t 的所有量子位设置为 $1/\sqrt{2}$, 对量子速度进行测量得到位置 \mathbf{x}_i^t , 量子粒子的局部最优位置设置为 $\mathbf{p}_i^t = \mathbf{x}_i^t$, $i = 1, 2, \dots, P$ 。

Step 02 对所有量子粒子进行适应度评价, 所有局部最优位置记作全局最优位置 \mathbf{p}_g^t 。

Step 03 根据式 (2.1.4)、式 (2.1.5) 和式 (2.1.6) 来更新每个量子粒子的量子速度 \mathbf{v}_i^{t+1} 和位置 \mathbf{x}_i^{t+1} 。

Step 04 对于每个量子粒子的新位置, 计算其适应度值。

Step 05 更新量子粒子的局部最优位置 \mathbf{p}_i^{t+1} 和全局最优位置 \mathbf{p}_g^{t+1} 。若 \mathbf{x}_i^{t+1} 优于 \mathbf{p}_i^t , 则 $\mathbf{p}_i^{t+1} = \mathbf{x}_i^{t+1}$; 否则 $\mathbf{p}_i^{t+1} = \mathbf{p}_i^t$ 。把第 $t+1$ 代中的最优的局部最优位置记作全局最优位置 \mathbf{p}_g^{t+1} 。

Step 06 如果进化并没有终止 (通常由预先设定的最大迭代次数决定), 则返回 **Step 03**; 否则, 算法终止。

2.1.2 单链编码的量子粒子群算法^[2]

由于 $\beta_{id}^t = \sqrt{1 - (\alpha_{id}^t)^2}$, 所以第 i 个量子粒子的量子速度被简化定义为

$$\mathbf{v}_i^t = [\alpha_{i1}^t, \alpha_{i2}^t, \dots, \alpha_{id}^t] = [v_{i1}^t, v_{i2}^t, \dots, v_{id}^t] \quad (2.1.7)$$

第 i 个量子粒子的第 d 个量子比特的量子旋转门更新过程简化如下:

$$v_{id}^{t+1} = |v_{id}^t \times \cos \theta_{id}^{t+1} - \sqrt{1 - (v_{id}^t)^2} \times \sin \theta_{id}^{t+1}| \quad (2.1.8)$$

如果量子旋转角 $\theta_{id}^{t+1} = 0$, 量子比特 \mathbf{v}_{id}^t 用量子非门以某种较小的概率进行更新, 这一过程可以简化为

$$v_{id}^{t+1} = \sqrt{1 - (v_{id}^t)^2} \quad (2.1.9)$$

在量子粒子群中, 有 P 个量子粒子在 D 维空间中搜寻最优位置。第 i 个量子粒子的位置为 $\mathbf{x}_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t]$, $1 \leq i \leq P$ 。第 i 个量子粒子的量子速度为 $\mathbf{v}_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t]$, 并且将第 i 个量子粒子迄今为止所搜到的局部最优位置记为 $\mathbf{p}_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t]$, $1 \leq i \leq P$ 。 $\mathbf{p}_g^t = [p_{g1}^t, p_{g2}^t, \dots, p_{gD}^t]$ 为整个量子粒子群迄今为止所搜到的全局最优位置。

第 i 个量子粒子的量子速度和位置的更新过程可以表示如下:

$$\theta_{id}^{t+1} = e_1(p_{id}^t - x_{id}^t) + e_2(p_{gd}^t - x_{id}^t) \quad (2.1.10)$$

$$v_{id}^{t+1} = \begin{cases} \sqrt{1 - (v_{id}^t)^2}, & \text{若 } (p_{id}^t = x_{id}^t = p_{gd}^t \text{ 且 } r < c_1) \\ \text{abs}(v_{id}^t \times \cos \theta_{id}^{t+1} - \sqrt{1 - (v_{id}^t)^2} \times \sin \theta_{id}^{t+1}), & \text{其他} \end{cases} \quad (2.1.11)$$

$$x_{id}^{t+1} = \begin{cases} 1, & r_{id}^{t+1} > (v_{id}^{t+1})^2 \\ 0, & r_{id}^{t+1} \leq (v_{id}^{t+1})^2 \end{cases} \quad (2.1.12)$$

其中, $1 \leq d \leq D$, r_{id}^{t+1} 为均匀分布在 $[0,1]$ 之间的均匀随机数, r 为均匀分布在 $[0,1]$ 之间的随机数, c_1 为变异概率, e_1 和 e_2 为两个常数, 分别决定了量子粒子的局部最优位置和整个量子粒子群的全局最优位置对该量子粒子的量子速度和位置的更新过程中的影响程度。

单链量子粒子群算法的步骤如下。

Step 01 初始化量子粒子群, 包括产生量子粒子的量子速度, 测量得到位置, 把最初的位置记作局部最优位置。

Step 02 对量子粒子进行适应度评价, 从而得到全局最优解。

Step 03 用式 (2.1.10)、式 (2.1.11) 和式 (2.1.12) 来更新量子粒子的量子速度和位置。

Step 04 计算每个量子粒子的新位置的适应度值。

Step 05 更新量子粒子的局部最优解, 同时找到全局的最优位置。

Step 06 如果进化并没有终止 (通常由预先设定的最大循环次数决定), 则返回 **Step 03**; 否则, 算法终止。

2.1.3 性能测试

为了测试本文所提出的量子粒子群算法（QPSO）的性能，选用经典的 Benchmark 函数对其进行测试。

1. Benchmark 函数测试

函数优化是人工智能算法的经典应用领域，也是对人工智能算法进行评价的常用手段之一。对于一些非线性、多模态、多目标的函数优化问题，其他传统方法难以求解，而用人工智能算法却可以很好地解决这些问题。为了验证量子粒子群算法的收敛速度和收敛到全局最优解的能力，选用了几个常用的 Benchmark 函数，对其性能进行测试。

参数设置如下：采用二进制编码方式，每个变量的编码长度为 15 位。

对于量子粒子群，参数设置如下： $e_1 = 0.06$ ， $e_2 = 0.03$ ， $c_1 = 0.1/D$ 。遗传算法（GA）、量子遗传算法（QGA）和粒子群算法（PSO）的参数设置参考文献[3]，其中更改遗传算法的变异概率为 0.02，设置的目的是使 Benchmark 函数 3 具有更好的效果。所有算法的群体规模设置为 20，仿真结果取 200 次试验的平均值。

算例 1:

$$f_1(\mathbf{x}) = \frac{1}{4000} \left(\sum_{i=1}^n (x_i - 100)^2 \right) - \left(\prod_{i=1}^n \cos \left(\frac{x_i - 100}{\sqrt{i}} \right) \right) \quad (2.1.13)$$

$+1, (-600 \leq x_i \leq 600, i=1,2)$

这是经典的 Griewank 函数。 $x_i (i=1,2)$ 在区间 $[-600, 600]$ 进行优化试验。这个函数的全局最小值为 0，其解为 $\mathbf{x}_{\text{opt}} = (x_1, x_2) = (100, 100)$ 。

从图 2.1.1 可以看出，尽管遗传算法、量子遗传算法和粒子群算法的收敛速度比较快，但明显容易陷入局部收敛；而量子粒子群算法的收敛精度相对于其他算法要高，在 100 代左右就得到了其他三种算法需要在 1000 代才能到达的收敛精度。

算例 2:

$$f_2(\mathbf{x}) = 2 \times 418.9829 + \sum_{i=1}^n -x_i \sin(\sqrt{|x_i|}), (-500 \leq x_i \leq 500, i=1,2) \quad (2.1.14)$$

这是经典的 Schwefel 函数，为多峰值多谷值的函数，该函数在距离全局最小值较远处有一个局部最小值，许多智能算法将陷入这个局部最小值，且全局最小值离优化区间的边界很近。 $x_i (i=1,2)$ 在区间 $[-500, 500]$ 进行优化试验。这个函数的全局最小值为 0，其解为 $\mathbf{x}_{\text{opt}} = (x_1, x_2) = (420.9867, 420.9867)$ 。

从图 2.1.2 可以看出，在此函数下，粒子群算法和遗传算法的收敛精度明显高于量子遗传算法但是收敛速度较慢，分别在 700 代和 800 代左右收敛。而量子粒子群算法在 150 代左右就已经基本收敛，并且收敛精度相对于经典的算法更高。在此函数下，若变异概率较小 (0.01)，则遗传性能较差，收敛性能与量子遗传算法相似。

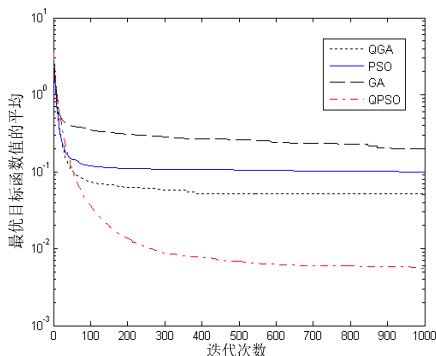


图 2.1.1 测试 Griewank 函数的收敛曲线

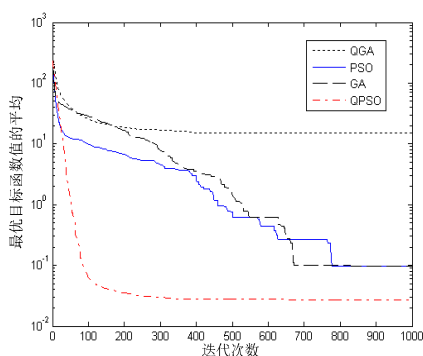


图 2.1.2 测试 Schwefel 函数的收敛曲线

算例 3:

$$f_3(\mathbf{x}) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.01(x_1^2 + x_2^2))^2}, (-100 \leq x_i \leq 100, i=1,2) \quad (2.1.15)$$

这是经典的 Schaffer 函数。 $x_i (i=1,2)$ 在区间 $[-100, 100]$ 进行优化试验。这个函数的全局最小值为 0，其解为 $\mathbf{x}_{\text{opt}} = (x_1, x_2) = (0, 0)$ 。

从图 2.1.3 可以看出,在此函数下,遗传算法、量子遗传算法、粒子群算法和量子粒子群算法具有相似的收敛速度和收敛精度,但是,量子粒子群算法的收敛精度相对于其他算法更高,这意味着量子粒子群算法收敛于全局最优解的能力更强,而其他三种算法陷入了局部收敛。

算例 4:

$$f_4(\mathbf{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), (-5.12 \leq x_i \leq 5.12, i=1,2) \quad (2.1.16)$$

这是经典的 Rastrigin 函数。该函数是一个具有大量局部最优点的复杂多峰函数,很容易陷入局部最优,用余弦项生成局部最优。 $x_i (i=1,2)$ 在区间 $[-5.12, 5.12]$ 进行优化试验。这个函数的全局最小值为 0, 其解为 $\mathbf{x}_{\text{opt}} = (x_1, x_2) = (0, 0)$ 。

从图 2.1.4 可以看出,遗传算法、量子遗传算法和粒子群算法具有相似的收敛速度与收敛精度,但是,其收敛精度相对于量子粒子群算法相当低,严重陷入了局部收敛。而量子粒子群算法,在 100 代左右就收敛到其他经典算法需要在 1000 代才能得到的精度,在 300 代左右已经收敛,并且收敛精度相当高。由此可见,所提出的算法具有很大的潜力。

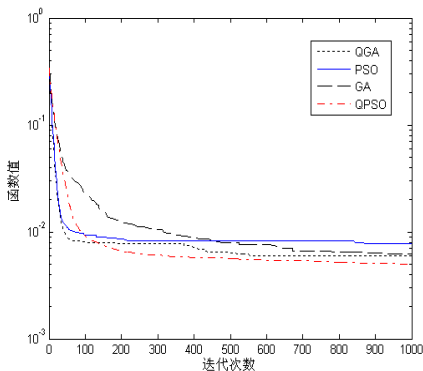


图 2.1.3 测试 Schaffer 函数的收敛曲线

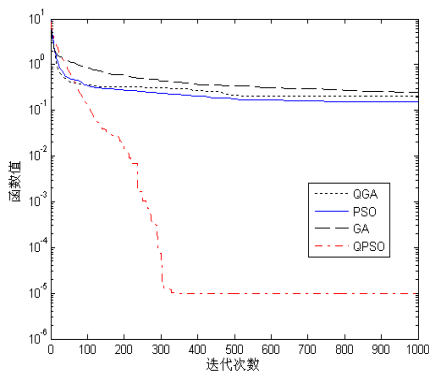


图 2.1.4 测试 Rastrigin 函数的收敛曲线

通过对 4 种 Benchmark 函数的测试,可以发现,遗传算法、量子遗传算法和粒子群算法的性能并不稳定,容易陷入局部收敛,而量子粒子群算法相对于这 3 种经典算法,具有很大的优势。

2.2 量子蜂群算法

尽管蜂群算法是为了解决连续优化问题提出的，但是本节通过量子编码和人工蜂群算法结合得到了双链编码^[4]和单链编码的离散量子蜂群算法，说明了使用量子编码的蜂群算法可有效解决离散优化问题，并使用测试函数说明了量子编码的蜂群算法具有性能上的优势。

2.2.1 双链编码的量子蜂群算法^[5]

量子蜜蜂的量子位置是由一串量子位表示的，一个量子位可表示为 $\mathbf{v}_{id}^t = [\alpha_{id}^t, \beta_{id}^t]^T, d=1, 2, \dots, D$ 。第 i 只量子蜜蜂的量子位置为

$$\mathbf{v}_i^t = \begin{bmatrix} \alpha_{i1}^t, \alpha_{i2}^t, \dots, \alpha_{iD}^t \\ \beta_{i1}^t, \beta_{i2}^t, \dots, \beta_{iD}^t \end{bmatrix} \quad (2.2.1)$$

其中， $|\alpha_{id}^t|^2 + |\beta_{id}^t|^2 = 1 (d=1, 2, \dots, D)$ ，将 α_{id}^t 和 β_{id}^t 定义为 $0 \leq \alpha_{id}^t \leq 1$ 、 $0 \leq \beta_{id}^t \leq 1$ 。

量子蜂群算法通过量子位置的更新来实现整个量子蜂群的演进。 θ_{id}^t 为第 t 次循环的一个量子旋转角，量子旋转门定义为 $U(\theta_{id}^t) = \begin{bmatrix} \cos \theta_{id}^t & -\sin \theta_{id}^t \\ \sin \theta_{id}^t & \cos \theta_{id}^t \end{bmatrix}$ ，量子比特 \mathbf{v}_{id}^t 用量子旋转门 $U(\theta_{id}^{t+1})$ 更新，第 i 只量子蜜蜂量子位置的第 d 个量子位的更新如下：

$$\mathbf{v}_{id}^{t+1} = \text{abs}(U(\theta_{id}^{t+1})\mathbf{v}_{id}^t) = \text{abs}\left(\begin{bmatrix} \cos \theta_{id}^{t+1} & -\sin \theta_{id}^{t+1} \\ \sin \theta_{id}^{t+1} & \cos \theta_{id}^{t+1} \end{bmatrix} \mathbf{v}_{id}^t\right) \quad (2.2.2)$$

如果量子旋转角 $\theta_{id}^{t+1} = 0$ ，量子比特 \mathbf{v}_{id}^t 用量子非门 \bar{N} 以某种较小的概率进行更新，即

$$\mathbf{v}_{id}^{t+1} = \bar{N}\mathbf{v}_{id}^t = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{v}_{id}^t \quad (2.2.3)$$

量子蜂群算法采用蜜蜂觅食行为的量子态演化。量子蜂群的前一半量子蜜蜂是由量子工蜂组成的，后一半量子蜜蜂是由量子观察蜂组成的。每一只量子工蜂或量子观察蜂都对应有一个食物源。每个量子蜜蜂，根据自己和其他量子蜜蜂的历史经验信息，在 D 维空间内搜索蜜源含量最佳的食物源位置。

在量子蜂群中有 h 只量子蜜蜂，对应 h 个食物源位置，每个食物源位置用 D 维取值 $\{0,1\}$ 的数字串代表。第 i 只量子蜜蜂的量子位置表示为 $\mathbf{v}_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$ ($i=1,2,\dots,h$)，量子位置的测量态为量子蜜蜂的位置，即搜索到的食物源的位置，则相应第 i 个食物源的位置可表示为 $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ ($i=1,2,\dots,h$)。在量子蜂群中，量子工蜂和量子观察蜂分别根据不同的演进规则寻找食物源，食物源位置的优劣由蜜源含量的大小来衡量。到目前为止第 i 只量子蜜蜂搜索到的最好位置（即局部最优位置）可以表示为 $\mathbf{p}_i = [p_{i1}, p_{i2}, \dots, p_{iD}]$ ($i=1,2,\dots,h$)。到现在为止整个量子蜂群所搜索到的全局最优位置（所有局部最优位置中的最优位置）可以表示为 $\mathbf{p}_g = [p_{g1}, p_{g2}, \dots, p_{gD}]$ 。量子工蜂的演进在其局部最优位置和整个蜂群当前所找到的全局最优位置共同指引下演进。但是，量子观察蜂由其局部最优位置、通过轮盘赌选择量子工蜂的局部最优位置和整个蜂群的全局最优位置指引量子位置的演进。

在每次循环中，第 i 只量子工蜂的第 d 维量子位的量子演进表示如下：

$$\theta_{id}^{t+1} = e_1(p_{id}' - x_{id}') + e_2(p_{gd}' - x_{id}') \quad (2.2.4)$$

$$\mathbf{v}_{id}^{t+1} = \begin{cases} \bar{N}\mathbf{v}_{id}', & \theta_{id}^{t+1}=0 \text{ 且 } \gamma_{id}^{t+1} < c_1 \\ \text{abs}[U(\theta_{id}^{t+1})\mathbf{v}_{id}'], & \text{其他} \end{cases} \quad (2.2.5)$$

其中， $i=1,2,\dots,h/2$ ， $d=1,2,\dots,D$ ； e_1 和 e_2 是两个影响因子，分别决定了局部最优位置和全局最优位置对量子旋转角的影响程度； γ_{id}^{t+1} 为 $[0,1]$ 之间均匀分布的随机数； c_1 是为量子工蜂在旋转角为 0 的量子位的变异概率，取值为 $[0,1/D]$ 之间的一个常数； $t+1$ 和 t 代表了循环次数（即迭代次数）。

通过观察量子工蜂的跳舞方式和持续时间长短等行为，第 i ($i=h/2+1, h/2+2, \dots, h$) 只量子观察蜂依概率选择量子工蜂，并根据其局部最优位置、所选择的量子工蜂的局部最优位置及食物源的全局最优位置进行位置

更新。使用 $fit(p'_j)$ 代表第 j 只量子工蜂局部最优位置的适应度，适应度函数是定义域内其值大于 0 的函数，越优秀的量子工蜂的适应度越高，被量子观察蜂选择的概率越大，则第 $j(j=1,2,\dots,h/2)$ 只量子工蜂被量子观察蜂轮盘赌选择的概率为

$$q_j^{t+1} = \frac{fit(p'_j)}{\sum_{k=1}^{h/2} fit(p'_k)} \quad (2.2.6)$$

在每次循环中，第 i 只量子观察蜂的量子位演进公式表示如下：

$$\theta_{id}^{t+1} = e_3(p'_{jd} - x_{id}^t) + e_4(p_{id}^t - x_{id}^t) + e_5(p_{gd}^t - x_{id}^t) \quad (2.2.7)$$

$$v_{id}^{t+1} = \begin{cases} \bar{N}v_{id}^t, & \text{若 } \theta_{id}^{t+1}=0 \text{ 且 } \gamma_{id}^{t+1} < c_2 \\ \text{abs}[U(\theta_{id}^{t+1})v_{id}^t], & \text{其他} \end{cases} \quad (2.2.8)$$

其中， $i=h/2+1, h/2+2, \dots, h$ ， $d=1,2,\dots,D$ ； e_3 、 e_4 和 e_5 是 3 个影响因子，分别决定了量子工蜂的局部最优位置、量子观察蜂的局部最优位置和整个量子蜂群的全局最优位置对量子旋转角的影响程度； c_2 为量子观察蜂在旋转角为 0 的量子位的变异概率，取值可选 $[0,1/D]$ 之间的常数。

第 i 只量子蜜蜂的食物源位置可以通过对量子位置各量子位的测量得到：

$$x_{id}^{t+1} = \begin{cases} 1, & \eta_{id}^{t+1} > (\alpha_{id}^{t+1})^2 \\ 0, & \eta_{id}^{t+1} \leq (\alpha_{id}^{t+1})^2 \end{cases} \quad (2.2.9)$$

其中， $\eta_{id}^{t+1} \in [0,1]$ 是满足均匀分布的随机数， $(\alpha_{id}^{t+1})^2$ 描述了量子位 v_{id}^{t+1} 出现“0”状态的概率。

根据上述分析和介绍，量子蜂群算法的迭代过程总结如下。

Step 01 初始量子蜂群，量子蜜蜂的量子位置中所有量子位均被初始化为 $1/\sqrt{2}$ 。测量量子位置的量子位得到食物源位置和量子蜜蜂的局部最优位置。

Step 02 根据蜜源含量函数计算量子蜜蜂所在位置的蜜源含量，确定量子蜂群的全局最优位置。

Step 03 用式 (2.2.4)、式 (2.2.5) 和式 (2.2.9) 来更新量子工蜂的量子位

置和食物源位置。

Step 04 量子观察蜂根据式 (2.2.6) 选择食物源。用式 (2.2.7)、式 (2.2.8) 和式 (2.2.9) 来更新量子观察蜂的量子位置和食物源位置。

Step 05 对于量子蜜蜂的新位置，计算蜜源含量。

Step 06 确定每只量子蜜蜂的局部最优位置，同时找到迄今为止的全局最优位置作为下一循环量子位的共同演进方向。

Step 07 如果演进并没有终止（通常由预先设定的最大循环次数决定），则返回 **Step 03**；否则，输出全局最优位置，算法终止。

2.2.2 单链编码的量子蜂群算法

量子位置可简化为 $\mathbf{v}_i = [\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iD}] = [v_{i1}, v_{i2}, \dots, v_{iD}]$ 。设 θ_{id}^{t+1} 为第 $t+1$ 次迭代的一个量子旋转角，量子比特 v_{id}^t 用模拟量子旋转门进行演进，第 i 只量子蜜蜂量子位置的第 d 个量子位的更新如下：

$$v_{id}^{t+1} = |v_{id}^t \cos(\theta_{id}^{t+1}) - \sqrt{1 - (v_{id}^t)^2} \sin(\theta_{id}^{t+1})| \quad (2.2.10)$$

量子非门可用如下公式模拟： $v_{id}^{t+1} = \sqrt{1 - (v_{id}^t)^2}$ 。

量子蜂群包含两种蜜蜂：量子工蜂和量子观察蜂。每一只量子蜜蜂在每一次循环中都会找到一个食物源，并且根据自己的历史经验和整个蜂群的经验，在位置空间中根据量子计算和群体智能演进规则找寻蜜源含量最优秀的位置^[6]。

在量子蜂群中，标号 $i=1, 2, \dots, \frac{h}{2}$ 的蜜蜂为量子工蜂，标号 $i=\frac{h}{2}+1, \frac{h}{2}+2, \dots, h$ 的蜜蜂为量子观察蜂。设 $\mathbf{v}_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t]$ ($i=1, 2, \dots, h$)，为第 i 只量子蜜蜂的量子位置，对量子位置进行测量得到量子蜜蜂的位置，也就是蜜源的位置，可表示为 $\mathbf{x}_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t]$ ($i=1, 2, \dots, h$)，其可用 D 维取值 $\{0, 1\}$ 的数字串代表。到目前为止第 i 只量子蜜蜂搜索到的局部最优位置可以表示为 $\mathbf{p}_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t]$ ($i=1, 2, \dots, h$)。到现在为止整个量子蜂群所搜索到的全局最

优位置可以表示为 $\mathbf{p}'_g = [p'_{g1}, p'_{g2}, \dots, p'_{gD}]$ 。

在第 $t+1$ 次迭代中, 第 i 只量子工蜂的第 d 维量子旋转角为

$$\theta_{id}^{t+1} = e_1(p'_{id} - x'_{id}) + e_2(p'_{gd} - x'_{id}) \quad (2.2.11)$$

其中, $i=1, 2, \dots, h/2$, $d=1, 2, \dots, D$; p'_{id} 和 p'_{gd} 对量子旋转角的影响程度分别由影响因子 e_1 和 e_2 决定; $t+1$ 和 t 代表了迭代次数。

第 i ($i=h/2+1, h/2+2, \dots, h$) 只量子观察蜂根据轮盘赌的方法选择量子工蜂参与量子旋转角的演进。设第 j 只量子工蜂局部最优位置的适应度为 $\hat{fit}(\mathbf{p}'_j)$, 构造适应度函数并使其值必大于 0, 则第 j ($j=1, 2, \dots, h/2$) 只量子工蜂被选择的概率为

$$q_j^{t+1} = \frac{\hat{fit}(\mathbf{p}'_j)}{\sum_{k=1}^{h/2} \hat{fit}(\mathbf{p}'_k)} \quad (2.2.12)$$

在第 $t+1$ 次迭代中, 第 i 只量子观察蜂的第 d 维量子旋转角为

$$\theta_{id}^{t+1} = e_3(p'_{jd} - x'_{id}) + e_4(p'_{id} - x'_{id}) + e_5(p'_{gd} - x'_{id}) \quad (2.2.13)$$

其中, $i=\frac{h}{2}+1, \frac{h}{2}+2, \dots, h$, $d=1, 2, \dots, D$; p'_{jd} 、 p'_{id} 和 p'_{gd} 对量子旋转角的影响程度分别由影响因子 e_3 、 e_4 和 e_5 决定。

第 i 只量子蜜蜂的第 d 位量子比特位置根据量子旋转角进行如下更新

$$v_{id}^{t+1} = \begin{cases} \sqrt{1-(v'_{id})^2}, & \text{if } \theta_{id}^{t+1}=0 \text{ and } \eta_{id}^{t+1} < c_1 \\ |v'_{id} \cos(\theta_{id}^{t+1}) - \sqrt{1-(v'_{id})^2} \sin(\theta_{id}^{t+1})|, & \text{else} \end{cases} \quad (2.2.14)$$

其中, $d=1, 2, \dots, D$; η_{id}^{t+1} 代表随机数, 它在 $[0, 1]$ 之间均匀分布; 而 c_1 代表当量子位的量子旋转角为 0 时对该量子位使用量子非门进行翻转操作的概率, 其值为 $[0, 1/D]$ 之间的常量, 根据不同的问题试验选择不同的值。

第 i 只量子蜜蜂的食物源位置可以对量子位置各量子位的测量得到:

$$x_{id}^{t+1} = \begin{cases} 1, & \lambda_{id}^{t+1} > (v_{id}^{t+1})^2 \\ 0, & \lambda_{id}^{t+1} \leq (v_{id}^{t+1})^2 \end{cases} \quad (2.2.15)$$

其中, $d = 1, 2, \dots, D$, λ_{id}^{t+1} 为 $[0, 1]$ 之间均匀分布的随机数, $(v_{id}^{t+1})^2$ 描述了量子位 v_{id}^{t+1} 出现“0”状态的概率。

2.2.3 性能测试

为了验证量子蜂群算法的收敛速度和搜索全局最优解的能力, 选用两个经典的 Benchmark 函数进行极小值测试。参数设置如下: 采用二进制编码方式, 每一个实变量的二进制编码长度为 15 位, 试验次数为 300, 仿真结果为 300 次运行的平均值。由于在求解工程优化问题时总希望计算量尽可能小, 因此所有智能算法的群体所含个体数为 20。在仿真过程中, 遗传算法 (GA)、量子遗传算法 (QGA) 和粒子群算法 (PSO) 的参数设置参考文献[3]和[7], 遗传算法的适应度函数设置为待优化函数的倒数。对于量子蜂群优化算法 (QBCO), 蜜源含量函数为待优化函数, 适应度函数设为蜜源含量函数的倒数。在仿真过程中, 设置参数如下: $e_1 = 0.06$, $e_2 = 0.03$, $e_3 = 0.03$, $e_4 = 0.06$, $e_5 = 0.01$, $c_1 = c_2 = 0.15/D$, D 为搜索空间维数。

算例 1 为经典的 Griewank 函数, 其表达式为

$$F_1(\mathbf{y}) = \frac{1}{4000} \left(\sum_{i=1}^2 (y_i - 100)^2 \right) - \left(\prod_{i=1}^2 \cos \left(\frac{y_i - 100}{\sqrt{i}} \right) \right) + 1, (-600 \leq y_i \leq 600, i = 1, 2) \quad (2.2.16)$$

从图 2.2.1 可以看出, 对于 Griewank 函数, 尽管遗传算法、量子遗传算法和粒子群算法的收敛速度比较快, 但明显容易陷入局部收敛; 而所提出的量子蜂群优化算法相对于其他算法有明显优势。

算例 2 为经典的 Schwefel 函数, 其表达式为

$$F_2(\mathbf{y}) = 2 \times 418.9829 - \sum_{i=1}^2 y_i \sin(\sqrt{|y_i|}), (-500 \leq y_i \leq 500, i = 1, 2) \quad (2.2.17)$$

该函数有一个距离全局最小值较远的局部最小值, 许多智能算法将陷入此局部最小值, 且全局最小值离优化区间的边界很近。从图 2.2.2 可以看出, 对于

Schwefel 函数，粒子群算法的收敛精度明显高于遗传算法和量子遗传算法，但是收敛速度慢；而量子蜂群算法的收敛速度和收敛精度都优于这三种算法。

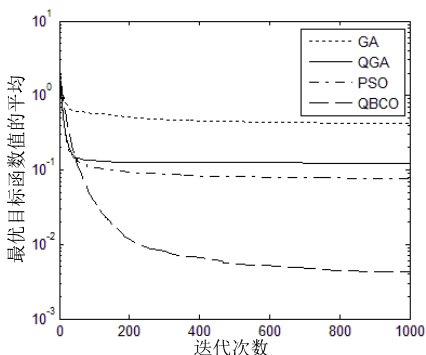


图 2.2.1 Griewank 函数的收敛曲线

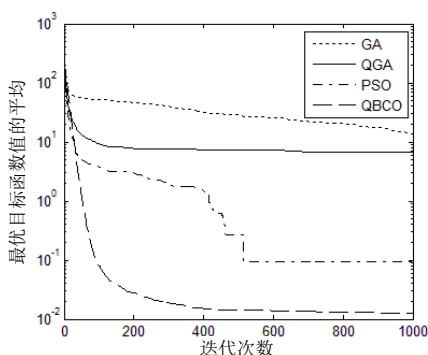


图 2.2.2 Schwefel 函数的收敛曲线

2.3 量子细菌觅食算法

细菌觅食算法是由 Passino 提出的一种高效的优化算法^[8]，它是受自然界生物在觅食过程中最大化每单位摄取的营养物质的生理机能启发而提出的^[9]，其模仿细菌在觅食过程中努力趋向于营养物质集中的地方、避开有毒有害物质存在的地方的能力。然而传统的细菌觅食算法仅能解决连续优化问题，本节设计了一种模拟量子旋转门实现的量子细菌觅食算法（Quantum-inspired Bacterial Foraging Algorithm, QBFA）求解离散优化问题。在 QBFA 中，同样模拟了细菌的趋向性、繁殖及驱散过程，算法核心操作利用量子旋转门的操作来完成，使得所获得的解向最优解演进。

2.3.1 双链编码的量子细菌觅食算法

在 QBFA 中，第 i 个量子细菌在第 l 次迁徙操作的第 q 次复制过程中的第 j 次趋向性操作时的量子位置可以表示为

$$\begin{aligned} \mathbf{v}_i(j, q, l) &= [\mathbf{v}_{i1}(j, q, l), \mathbf{v}_{i2}(j, q, l), \dots, \mathbf{v}_{iD}(j, q, l)] \\ &= \begin{bmatrix} \alpha_{i1}(j, q, l) & \alpha_{i2}(j, q, l) & \dots & \alpha_{iD}(j, q, l) \\ \beta_{i1}(j, q, l) & \beta_{i2}(j, q, l) & \dots & \beta_{iD}(j, q, l) \end{bmatrix} \end{aligned} \quad (2.3.1)$$

其中, $|\alpha_{id}(j, q, l)|^2 + |\beta_{id}(j, q, l)|^2 = 1$, $0 \leq \alpha_{id}(j, q, l) \leq 1$, $0 \leq \beta_{id}(j, q, l) \leq 1$, $(d=1, 2, \dots, D)$ 。 $|\alpha_{id}(j, q, l)|^2$ 代表了量子比特选择状态为“0”的概率^[10], D 为所需优化问题的最大维数。

QBFA 是一个新颖的群优化算法, 在一个 D 维优化问题中, 量子细菌总数为 S 。第 i 个量子细菌的位置可表示为 $\mathbf{x}_i(j, q, l) = [x_{i1}(j, q, l), x_{i2}(j, q, l), \dots, x_{iD}(j, q, l)]$ ($i=1, 2, \dots, S$), 它是优化问题的一个潜在解。第 i 个量子细菌到目前为止所找到的最优秀量子位置为局部最优量子位置 $\mathbf{b}_i(j, q, l) = [b_{i1}(j, q, l), b_{i2}(j, q, l), \dots, b_{iD}(j, q, l)]$ ($i=1, 2, \dots, S$), 整个量子细菌种群到目前为止所找到的最优秀量子位置为全局最优量子位置 $\mathbf{b}_g(j, q, l) = [b_{g1}(j, q, l), b_{g2}(j, q, l), \dots, b_{gD}(j, q, l)]$ 。每只量子细菌的量子位置的更新过程由其局部最优量子位置和全局最优量子位置共同指导完成。

在一次旋转移动过程中, 量子细菌利用量子旋转门操作来完成趋向性操作。第 i 个量子细菌的第 d 维量子旋转角及量子位置采取如下的更新方法:

$$\theta_{id}(j+1, q, l) = e_1[b_{id}(j, q, l) - x_{id}(j, q, l)] + e_2[b_{gd}(j, q, l) - x_{id}(j, q, l)] \quad (2.3.2)$$

$$\mathbf{v}_{id}(j+1, q, l) = \begin{cases} \mathbf{v}_{id}(j, q, l), & \theta_{id}(j+1, q, l) = 0 \\ \text{abs}[\mathbf{G}(\theta_{id}(j+1, q, l))\mathbf{v}_{id}(j, q, l)], & \text{其他} \end{cases} \quad (2.3.3)$$

其中, $\text{abs}(\cdot)$ 代表取绝对值运算, $\theta_{id}(j+1, q, l)$ 代表量子旋转角, e_1 和 e_2 分别代表 \mathbf{b}_i 和 \mathbf{b}_g 对量子旋转角演化的影响程度, 随着量子细菌的移动, 它释放信号来吸引其他的量子细菌向其趋近。

$\mathbf{G}(\theta_{id}(j+1, q, l))$ 代表量子旋转门, 可以具体表示为

$$\mathbf{G}[\theta_{id}(j+1, q, l)] = \begin{bmatrix} \cos[\theta_{id}(j+1, q, l)] & -\sin[\theta_{id}(j+1, q, l)] \\ \sin[\theta_{id}(j+1, q, l)] & \cos[\theta_{id}(j+1, q, l)] \end{bmatrix} \quad (2.3.4)$$

第 i 只量子细菌的位置的第 d 维位置通过对与其相对应的量子位置进行测

量来获得，方法如下：

$$x_{id}(j+1, q, l) \begin{cases} 1, & \gamma_{id}(j+1, q, l) > (a_{id}(j+1, q, l))^2 \\ 0, & \gamma_{id}(j+1, q, l) \leq (a_{id}(j+1, q, l))^2 \end{cases} \quad (2.3.5)$$

其中， $d=1, 2, \dots, D$ ， $\gamma_{id}(j+1, q, l) \in [0, 1]$ 是一个均匀随机数。

第 i 个量子细菌在位置 $\mathbf{x}_i(j, q, l)$ 的适应度值用 $f[\mathbf{x}_i(j, q, l)]$ 来计算，并将最大适应度值 f_{\max} 定义为最优适应度值。第 i 个量子细菌的健康值由式 (2.3.6) 来计算

$$f_{\text{health}}^i(j, q, l) = f[\mathbf{b}_i(j, q, l)] \quad (2.3.6)$$

其中， $f_{\text{health}}^i(j, q, l)$ 表示第 i 个量子细菌的健康值。

对于一个最大值优化问题，量子细菌的健康值越大，量子细菌越健康。为模拟量子细菌的繁殖过程，将所有的量子细菌按照降序排列，假设细菌总数为偶数。在前一半量子细菌中，每一个量子细菌进行无变异分裂，分裂为两个与父代相同的量子细菌；后一半量子细菌被删除和遗弃。这样的操作可以保证在进化过程中量子细菌总数不变。

基于上述的分析与介绍，量子细菌觅食算法的流程可以归纳如下。

Step 01 初始化种群规模 S 、量子趋向性操作次数 N_c 、复制操作次数 N_{re} 、迁徙操作过程次数 N_{ed} 及其概率 P_{ed} 。将量子细菌的所有量子位置的量子比特位初始化为 $1/\sqrt{2}$ ，并将量子位置测量为位置。计算适应度，初始化每只量子细菌的局部最优位置和细菌群体的全局最优位置。

Step 02 迁徙操作循环： $l = l + 1, q = 0$ 。

Step 03 繁殖操作循环： $q = q + 1, j = 0$ 。

Step 04 量子趋向性操作循环： $j = j + 1$ 。

[a] 依次对量子细菌 i 实施量子趋向性操作，其中， $i = 1, 2, \dots, S$ 。

[b] 旋转：对量子细菌 i 产生量子旋转角 $\theta_{id}(j+1, q, l)$ 。

[c] 移动：对于量子细菌 i ，根据式 (2.3.3) 和式 (2.3.5) 更新其量子位置和位置。

[d] 如果 $i \neq S$ ，进入步骤[b]处理下一个量子细菌。

[e] 计算适应度函数 $f(\mathbf{x}_i(j+1, q, l))$ 。如果 $f(\mathbf{x}_i(j+1, q, l))$ 优于 $f(\mathbf{b}_i(j, q, l))$ ，则 $\mathbf{b}_i(j+1, q, l) = \mathbf{x}_i(j+1, q, l)$ ；否则 $\mathbf{b}_i(j+1, q, l) = \mathbf{b}_i(j, q, l)$ 。如果 $\mathbf{b}_i(j+1, q, l)$ 优于 $f(\mathbf{b}_g(j, q, l))$ ，则 $\mathbf{b}_g(j+1, q, l) = \mathbf{b}_i(j+1, q, l)$ ， $\mathbf{b}_g(j, q, l) = \mathbf{b}_i(j+1, q, l)$ ；否则 $\mathbf{b}_g(j+1, q, l) = \mathbf{b}_g(j, q, l)$ 。

Step 05 如果 $j < N_c$ ，进入 **Step 04** 继续进行趋向性操作。

Step 06 复制操作：

[a] 在设定 q 和 l 的情况下，对于每一个量子细菌 $i(i=1, 2, \dots, S)$ ，计算其健康值 $f_{\text{health}}^i(N_c, q, l)$ 的大小，并对所有量子细菌的健康值以降序排列。

[b] 具有较低健康值的 $S/2$ 个量子细菌消亡，具有较高健康值的 $S/2$ 个量子细菌进行分裂。

Step 07 如果 $q < N_{re}$ ，则进入 **Step 03**，继续完成下一代复制操作，开始进行新一轮的量子趋向性操作。

Step 08 迁徙操作：除去最健康的量子细菌，对于每一只量子细菌，以概率 P_{ed} 进行变异操作。如果对某一量子细菌进行变异，即随机设置该量子细菌的位置，并将其量子位置的量子比特位设置为 $1/\sqrt{2}$ 。如果 $l < N_{ed}$ ，则进入 **Step 02**；否则，终止算法。

为了便于分析运算开销，令 t 代表 QBFA 算法的当前迭代次数，具体表达式为 $t = N_c \cdot (q-1) + N_c \cdot N_{re} \cdot (l-1) + j$ 。

2.3.2 量子细菌觅食算法的收敛性分析

定义 1: 量子细菌觅食算法的种群规模为 S ， X 为细菌搜索空间，它包含所

有可能的位置。定义 $X^t = \{x_1^t, x_2^t, \dots, x_S^t, b_1^t, b_2^t, \dots, b_S^t\}$ 为第 t 次迭代的位置和局部最优位置集合, f 为搜索空间 X 的目标函数, f^* 为全局最优目标函数值, 令

$$Q = \{\bar{X} \mid f(\bar{X}) = \max_{x_i^t, b_i^t \in X^t} \{f(x_i^t), f(b_i^t), i = 1, \dots, S\}\} \quad (2.3.7)$$

其中, Q 被称为种群 X^t 的满意集合。

定义 2: 令 $f_t = \max_{x_i^t \in X^t, b_i^t \in X^t} \{f(x_i^t), f(b_i^t), i = 1, 2, \dots, S\}$ 。对于任意初始分布, 如果下式成立:

$$\lim_{t \rightarrow \infty} P\{f_t = f^*\} = 1 \quad (2.3.8)$$

其中, P 表示概率, 那么这个算法将以概率 1 收敛^[11]。

这个收敛性的定义意味着如果算法运行足够的迭代次数, 那么此位置集合中包含最优位置的概率将趋近于 1。

引理 1: QBFA 位置集合 $\{X^t, t > 0\}$ 是一个有限齐次马可科夫链。

证明: QBFA 的状态转移过程是在有限的空间上进行的。因此, 位置集合是有限的, 因为

$$X^{t+1} = T(X^t) = T_C \cdot T_R \cdot T_E(X^t) \quad (2.3.9)$$

其中, T_C 、 T_R 和 T_E 分别表示量子趋向算子、复制算子和驱散消除算子。值得注意的是这三种算子与迭代次数 t 无关, 而 X^{t+1} 只与 X^t 相关。因此 $\{X^t, t > 0\}$ 是一个有限齐次马可科夫链。

引理 2: QBFA 的马可科夫链 Q 是单调递增的, 也就是说, $\forall t > 0$, $\exists f(X^{t+1}) \geq f(X^t)$ 。

证明: 结论是显然的, 因为 QBFA 算法使用了最优解保留策略, 随着迭代次数的增加, 种群中最优目标函数值是非减的。

定理 1: QBFA 算法是收敛的。

证明: 将种群规模为 S 的种群看作状态空间的一个状态 $Z = X^S$, 其中每个具体的元素为 X 中的一个位置。令 S_n 表示状态空间 Z 的状态数目。其中 $z_i \in Z$

$(i=1,2,\dots,S_n)$ 是一个确定的状态 $z_i = \{x_1^i, x_2^i, \dots, x_S^i, b_1^i, b_2^i, \dots, b_S^i\}$ 。 X_i^t 可看作第 t 次迭代在状态 z_i 的一个随机变量(即种群)。令 $P_{ik}(t)$ 表示从状态 X_i^t 到状态 X_k^{t+1} 的转移概率, 即

$$P_{ik}(t) = P\{X_k^{t+1} | X_i^t\} \quad (2.3.10)$$

设 $I = \{i | z_i \cap z^* \neq \emptyset\}$, 其中 z^* 表示全局最优位置。首先应该考虑两种特殊情况:

(1) 如果 $i \in I, k \notin I$, 根据引理 1, 那么 $P_{ik}(t) = 0$ 。

(2) 如果 $i \notin I, k \in I$, 根据引理 2, 那么 $f(X_k^{t+1}) \geq f(X_i^t)$, 因此 $P_{ik}(t) > 0$ 。

接下来考虑一般情况: 令 $P_i(t) = P\{X^t = z_i\}$, 其中 P 表示概率, 且 $P_t = \sum_{i \in I} P_i(t)$ 。根据引理 1, P_{t+1} 可以被记作

$$P_{t+1} = \sum_{z_i \in Z} \sum_{k \notin I} P_i(t) P_{ik}(t) = \sum_{i \in I} \sum_{k \notin I} P_i(t) P_{ik}(t) + \sum_{i \notin I} \sum_{k \in I} P_i(t) P_{ik}(t) \quad (2.3.11)$$

因为

$$\sum_{i \notin I} \sum_{k \notin I} P_i(t) P_{ik}(t) + \sum_{i \notin I} \sum_{k \in I} P_i(t) P_{ik}(t) = \sum_{i \notin I} P_i(t) = P_t \quad (2.3.12)$$

由此可得

$$\sum_{i \notin I} \sum_{k \notin I} P_i(t) P_{ik}(t) = P_t - \sum_{i \notin I} \sum_{k \in I} P_i(t) P_{ik}(t) \quad (2.3.13)$$

因此

$$\lim_{t \rightarrow \infty} P_t = 0 \quad (2.3.14)$$

根据式 (2.3.12)、式 (2.3.13) 和式 (2.3.14), 可以得到 $0 \leq P_{t+1} < \sum_{i \in I} \sum_{k \notin I} P_i(t) P_{ik}(t) + P_t = P_t$, 则 $\lim_{t \rightarrow \infty} P_t = 0$ 。

因为 $\lim_{t \rightarrow \infty} P\{f_t = f^*\} = 1 - \lim_{t \rightarrow \infty} \sum_{i \notin I} P_i(t) = 1 - \lim_{t \rightarrow \infty} P_t$, 所以可以得出 $\lim_{t \rightarrow \infty} P\{f_t = f^*\} = 1$ 。上述过程说明了 QBFA 能够收敛到全局最优解。

2.3.3 性能测试

利用两个基准函数的最小值测试所提出的 QBFA，并将所得结果和遗传算法 (GA)、量子遗传算法 (QGA) 及免疫克隆选择算法 (Immune Clone Selection Algorithm, ICSEA) 进行了比较。为便于比较，这 4 种智能算法的种群规模及最大迭代次数设置相同。种群规模设置为 20。GA 参数设置：交叉概率为 0.8，变异概率为 0.02^[3]；QGA 参数设置：量子旋转角从初始代 0.1π 线性递减至最后一代的 0.005π ^[3]。ICSEA 参数设置：记忆集合个体数为 6，变异概率为 0.1，克隆规模控制参数设为 $20^{[12]}$ 。QBFA 参数设置情况如下： $N_c=500$ ， $N_{ed}=2$ ， $N_{re}=1$ ， $e_1=0.06$ ， $e_2=0.015$ ， $p_{ed}=0.1$ ，最大迭代次数为 $N_c \cdot N_{re} \cdot N_{ed}=1000$ 。

所选取的基准函数为 Griewank 函数和 Rastrigin 函数，其公式分别为

$$F_1(\mathbf{x}) = \frac{1}{4000} \left(\sum_{i=1}^n (x_i - 100)^2 \right) - \left(\prod_{i=1}^n \cos \left(\frac{x_i - 100}{\sqrt{i}} \right) \right) + 1, (-600 \leq x_i \leq 600, i=1, 2, \dots, M) \quad (2.3.15)$$

$$F_2(\mathbf{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), (-5.12 \leq x_i \leq 5.12, i=1, 2, \dots, M) \quad (2.3.16)$$

在下述仿真过程中，每个变量使用 30 个二进制比特进行编码，对于所有基准函数的参数 M 和 n 都设置为 2。所有的仿真结果是 200 次实验仿真结果的平均值，两个基准函数的最优目标函数值的平均值和迭代次数关系曲线如图 2.3.1 和图 2.3.2 所示。

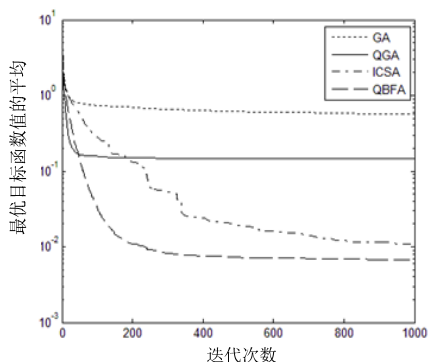


图 2.3.1 Griewank 函数的收敛曲线

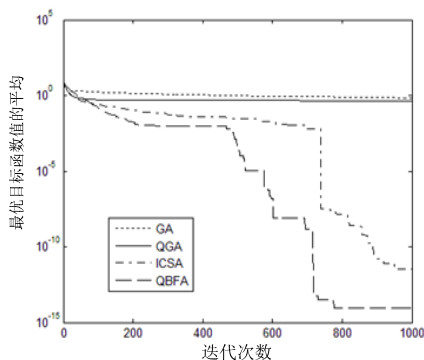


图 2.3.2 Rastrigin 函数的收敛曲线

从图 2.3.1 和图 2.3.2 中可以看出, ICSA、QGA 和 GA 收敛性能较差, 而所提出的 QBFA 无论在收敛速度还是收敛精度上都是最优的。仿真结果验证了所提出算法的优越性与高效性。

2.4 小结

本章主要讨论了 3 种解决离散优化问题的量子群智能算法——量子粒子群算法、量子蜂群算法和量子细菌觅食算法, 并且将它们与遗传算法、量子遗传算法和粒子群算法等典型智能计算算法进行了仿真和比较。由 Benchmark 函数测试的仿真结果可知, 量子群智能算法与经典智能算法相比, 具有更快的收敛速度和更高的收敛精度, 从而验证了其有效性和优越性。

参考文献

- [1] Gao Hongyuan, Diao Ming. Quantum particle swarm optimization for MC-CDMA multiuser detection. 2009 International Conference on Artificial Intelligence and Computational Intelligence. 2009, 2: 132-136.
- [2] Gao Hongyuan, Cao Jinlong, Diao Ming. A simple quantum-inspired particle swarm optimization and its application. Information Technology Journal. 2011,10(12): 2315- 2321.
- [3] Zhao Zhijin, Peng Zhen, Zheng Shilian, Shang Junna. Cognitive radio spectrum allocation using evolutionary algorithms. IEEE Transactions on Wireless Communications. 2009, 8(9):4421-4425.
- [4] Gao Hongyuan, Liu Yuqi, Diao Ming. Robust multi-user detection based on quantum bee colony optimization. International Journal of Innovative Computing and Applications .2011,3(3):160-168.

- [5] 高洪元, 曹金龙. 量子蜂群算法及其在认知频谱分配中的应用. 中南大学学报 (自然科学版), 2012, 43 (12): 4744-4749.
- [6] Gao Hongyuan, Li Chenwan, Cui Wen. A simple quantum-inspired bee colony algorithm for discrete optimisation problems. *International Journal of Computer Applications in Technology*. 2013, 46(3): 244-251.
- [7] 赵知劲, 彭振, 郑仕链, 等. 基于量子遗传算法的认知无线电频谱分配. 物理学报, 2009, 58 (2): 1358-1363.
- [8] Passino K M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*. 2002, 22(3): 52-67.
- [9] Das S, Dasgupta S, Biswas A, Abraham A, Konar A. On stability of the chemotactic dynamics in bacterial-foraging optimization algorithm. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*. 2009, 39(3): 670-679.
- [10] Gao Hongyuan, Cui Wen, Li Chenwan. A quantum bacterial foraging optimisation algorithm and its application in spectrum sensing. *International Journal of Modelling, Identification and Control*. 2013, 18(3): 234-242.
- [11] Jiao Licheng, Li Yangyang, Gong Maoguo, Zheng Xiangrong. Quantum-inspired immune clonal algorithm for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, 1234-1253.
- [12] 柴争义, 刘芳. 基于免疫克隆选择优化的认知无线网络频谱分配. 通信学报, 2010, 31 (11): 92-100.

第 3 章

用于连续优化问题的量子群智能计算

在过去的十几年里，群集智能算法和理论已经得到人们的关注，产生了众多解决连续优化问题的优化算法。典型的群智能优化算法，如粒子群优化算法、细菌觅食算法、蜂群算法和混合蛙跳算法等，现被广泛关注和研究改进，并成功运用在一些具体优化问题上。针对不同种类的连续优化问题，这些算法寻优各有特点。现有的群集智能算法与进化算法一样在工程应用和理论计算时面临着收敛速度和收敛性能之间的矛盾。在求解高维优化问题时，为了避免早熟收敛，要求有大的群体规模和迭代次数，这在实时实现上需要付出很大代价；且一些算法虽经过理论证明是收敛的，但收敛性并不强，都有适用的范围，在面对工程中复杂难解的多维搜索问题时，算法性能恶化严重。因此，针对不同的工程和理论问题，改进群集智能会提供更多的新方法。

本章针对连续优化问题的求解，把量子优化计算和混合蛙跳算法、文化蛙跳算法及细菌觅食算法结合，设计了三种连续优化算法，并使用测试函数验证了其有效性。

3.1 量子蛙跳算法

混合蛙跳算法^[1,2]根据青蛙家族的跳跃行为进行最优值的求解,在解决连续优化问题时具有较快的收敛速度。其基本思想来源于文化基因传承,显著特点是通过局部搜索与全局信息混合的协同搜索策略来寻找到最优解^[3]。经过大量仿真测试表明,在解决高维、多局部极值等函数问题时,混合蛙跳算法具有一定的优越性,是一种行之有效的优化技术。该算法具有概念简单、参数少、计算速度快、易于实现、全局寻优能力强等特点,目前在函数优化、水资源网络分配问题和成品油管网优化等方面得到了广泛应用,并取得了良好的效果。最为重要的是,与其他智能算法相比较,混合蛙跳算法具有比较少的控制参数这一优势。但是,蛙跳算法具有容易陷入局部收敛的不足。因此本节考虑引入量子编码,使得蛙跳算法量子化,设计量子蛙跳算法(QSFLA)。

3.1.1 混合蛙跳算法

假设青蛙种群的规模为 P ,其中第 i 只青蛙个体在 M 维空间的坐标表示为 $\mathbf{X}_i = [X_{i1}, X_{i2}, \dots, X_{iM}]$,每只青蛙个体的适应度为 $f(\mathbf{X}_i)$,按照适应度的降序排列。然后将所有青蛙个体分成 m 个族群,每个族群包含 n 个青蛙个体。分配过程的策略如下:第1只青蛙进入第1个族群,第2只青蛙进入第2个族群,……,第 m 只青蛙进入第 m 个族群。然后,第 $m+1$ 只青蛙又进入第1个族群,依此类推,直到全部青蛙划分完毕。在每个族群中,具有最优适应度和最差适应度的青蛙位置分别为 $\mathbf{X}_b = [X_{b1}, X_{b2}, \dots, X_{bM}]$ 和 $\mathbf{X}_w = [X_{w1}, X_{w2}, \dots, X_{wM}]$ 。全体青蛙中具有全局最优适应度的青蛙位置为 $\mathbf{X}_g = [X_{g1}, X_{g2}, \dots, X_{gM}]$ 。在族群内的一次迭代过程中,对于族群最差适应度对应的青蛙位置 \mathbf{X}_w 的第 j 维分量按照如下策略进行更新

$$D_j = \text{rand} \cdot (X_{bj} - X_{wj}) \quad (3.1.1)$$

$$X_{wj}^{\text{new}} = X_{wj} + D_j \quad (3.1.2)$$

其中, D_j 表示更新时分量 j 移动的距离, 初始情况下随机生成 $D_j \in [d_j^{\min}, d_j^{\max}]$, 其中 d_j^{\min} 和 d_j^{\max} 分别表示最小和最大移动步长。rand 为 $[0,1]$ 间的均匀随机数。如果 X_w^{new} 的适应度值没有改进, 则将公式 (3.1.1) 中的 X_{bj} 替换为 X_{gj} 使得青蛙更快地向全局最优位置移动, 并且按照如下方式更新

$$D_j = \text{rand} \cdot (X_{gj} - X_{wj}) \quad (3.1.3)$$

$$X_{wj}^{\text{new}} = X_{wj} + D_j \quad (3.1.4)$$

如果 X_w^{new} 的适应度仍然没有改进, 则在定义域区间内随机产生新的青蛙位置 X_w^{new} , 这样便完成了族群内的一次迭代。重复上述更新过程直到达到族群的最大迭代次数。对所有族群执行上述相同的更新过程, 这样便完成了一个进化过程。完成该进化过程之后, 将所有青蛙混合, 重新计算所有青蛙的当前位置的适应度并且按照适应度的降序方式排列青蛙, 然后进行下一个全局迭代过程。

通过上述分析与介绍, 基于混合蛙跳算法解决优化问题的流程可以归纳如下。

Step 01 参数初始化。赋予青蛙种群总数 p 、青蛙族群数 m 、每个族群内的青蛙数 n 以适当的数值。还需要设置族群内的最大迭代次数 gen 及整个群体的最大演进代数 Gen 。在定义域内随机产生青蛙的初始位置。

Step 02 计算每个青蛙的适应度值, 并按照适应度值的降序方式排列青蛙。具有全局最优适应度的青蛙位置为 X_g 。

Step 03 将全部群体分成 m 个族群。划分规则: 第 1 只青蛙进入第 1 个族群, 第 2 只青蛙进入第 2 个族群, ……第 m 只青蛙进入第 m 个族群。然后, 第 $m+1$ 只青蛙又进入第 1 个族群, 依此类推, 直到全部青蛙划分完毕。族群内最好适应度值和最差适应度值的青蛙位置分别为 X_b 和 X_w 。

Step 04 令 $i_m=0$, i_m 为族群的序号。

Step 05 $i_m = i_m + 1$, $i_{gen}=0$, i_{gen} 为族群内的迭代次数序号。

Step 06 令 $i_{gen} = i_{gen} + 1$ 。

Step 07 在族群 i_m 内, 按照混合蛙跳算法的更新策略更新最差适应度值的青蛙位置 X_w 。首先, 按照如下方程更新 X_w : $D_j = \text{rand} \cdot (X_{bj} - X_{wj})$, $X_{wj}^{\text{new}} = X_{wj} + D_j, j=1,2,\dots,M$ 。

如果新的青蛙量子位置的适应度值没有改进, 则在更新策略中使用 X_g 代替 X_b , 使其更快地向全局最优解收敛, $D_j = \text{rand} \cdot (X_{gj} - X_{wj})$, $X_{wj}^{\text{new}} = X_{wj} + D_j, j=1,2,\dots,M$ 。如果适应度值仍然没有改进, 那么在定义域区间随机产生新的青蛙位置 X_w^{new} 。

这样便完成了族群 i_m 的一次循环, 更新该族群内的 X_b 、 X_w 和 X_g 。

Step 08 如果 $i_{gen} < gen$, 则进入 **Step 06**; 否则, 执行 **Step 09**。

Step 09 如果 $i_m < m$, 则进入 **Step 05**。

Step 10 测试终止条件。如果达到最大进化代数, 则输出最终 X_g 及其对应的最优值, 终止算法; 否则, 按照适应度降序的方式排列青蛙, 进入 **Step 03**。

3.1.2 量子蛙跳算法^[4]

混合蛙跳算法是基于特定青蛙个体及全体青蛙个体的信息交换过程来进化的一种群体智能方法。但是混合蛙跳算法并不改变青蛙个体的信息, 也不会综合全部个体的主流思想, 这样会限制算法的寻优能力, 为此提出量子蛙跳算法 (QSFLA), 使用单链编码, 模拟量子旋转门演化其量子位置。

第 i 只青蛙的量子位置可以表示为 $x_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$, 第 i 只青蛙的量子位置的第 j 个量子位位置模拟量子旋转门进行更新

$$x_{ij}^{\text{new}} = |x_{ij} \cos \theta_j - \sqrt{1 - (x_{ij})^2} \sin \theta_j| \quad (3.1.5)$$

其中, $0 \leq x_{ij} \leq 1 (j=1,2,\dots,M)$, θ_j 为量子旋转角。

用 $\bar{x}_i = [\bar{x}_{i1}, \bar{x}_{i2}, \dots, \bar{x}_{iM}]$ 表示第 i 只青蛙的位置, 即为量子位置 $x_i =$

$[x_{i1}, x_{i2}, \dots, x_{iM}]$ 的映射位置, 其中 $0 \leq x_{ij} \leq 1$, $\bar{x}_{ij} = \bar{x}_j^{\min} + x_{ij}(\bar{x}_j^{\max} - \bar{x}_j^{\min})$, $\bar{x}_j^{\min} \leq \bar{x}_{ij} \leq \bar{x}_j^{\max}$ 。 \bar{x}_j^{\min} 是第 j 维变量的下限, \bar{x}_j^{\max} 是第 j 维变量的上限。第 i 只青蛙的量子位置适应度由其位置的适应度值 $f(\bar{\mathbf{x}}_i)$ 来度量。

在一个 M 维优化问题中, 在量子蛙跳算法的开始阶段, 在量子位的定义区间随机产生 p 只青蛙的量子位置。第 i 只青蛙的量子位置表示为 $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iM}]$ 。在一次进化中, 首先将所有的青蛙按照它们适应度值的降序排列。然后将全部青蛙分为 m 个族群, 每个族群包含 n 只青蛙 (则 $p = m \times n$)。划分过程中, 第 1 只青蛙进入第 1 个族群, 第 2 只青蛙进入第 2 个族群, \dots , 第 m 只青蛙进入第 m 个族群。然后, 第 $m+1$ 只青蛙又进入第 1 个族群, 依此类推, 直到全部青蛙划分完毕。在每一代中, 每个族群中具有最好适应度值和最差适应度值的青蛙量子位置分别为 $\mathbf{x}_b = [x_{b1}, x_{b2}, \dots, x_{bM}]$ 和 $\mathbf{x}_w = [x_{w1}, x_{w2}, \dots, x_{wM}]$ 。全体青蛙中具有全局最好适应度值的青蛙量子位置为 $\mathbf{x}_g = [x_{g1}, x_{g2}, \dots, x_{gM}]$ 。

上述过程将青蛙种群划分为若干个族群, 每个族群包括一个子群。每个青蛙的跳跃方程由下述策略决定:

按照如下方式更新具有最差适应度值的青蛙的量子位置 \mathbf{x}_w

$$\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{bj}) + \text{randn} \cdot (x_{wj} - z_j) \quad (3.1.6)$$

$$x_{wj}^{\text{new}} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}| \quad (3.1.7)$$

其中, rand 为 $[0,1]$ 之间的均匀随机数, randn 为满足均值为 0、方差为 1 的高斯分布的随机数。若 z 为第 i_m 个族群内全部青蛙的量子位置的平均值

$$z = [z_1, z_2, \dots, z_M] \quad (3.1.8)$$

其中, $z_j = \frac{1}{n} \sum_{i=1}^n x_{[m(i-1)+i_m], j}$, $j = 1, 2, \dots, M$ 。

如果新的量子位置 $\mathbf{x}_w^{\text{new}}$ 的适应度值没有改进, 则按照下述方式进行更新:

$$\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{gj}) + \text{randn} \cdot (x_{wj} - z_j) \quad (3.1.9)$$

$$x_{wj}^{\text{new}} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}| \quad (3.1.10)$$

如果适应度值仍然没有改进，那么在定义域区间随机产生新的青蛙量子位置 $\mathbf{x}_w^{\text{new}}$ 。

重复上述过程直到达到族群内部的最大迭代次数。当全部族群完成了内部进化之后，将青蛙种群重新进行混合分组，进入下一代进化，直到达到预先设定的最大进化代数。

基于上述分析与介绍，量子蛙跳算法的流程可以归纳如下。

Step 01 参数初始化。赋予青蛙群体总数 p 、青蛙族群数 m 、每个族群内的青蛙数 n 以适当的数值，因此可以得到 $p = m \times n$ 。同样还需要设置族群内的最大迭代次数 gen 及整个群体的最大演进代数 Gen 。在量子位置定义域内随机产生青蛙的初始量子位置。

Step 02 把量子位置映射到位置，计算每个青蛙的适应度值，并按照适应度值的降序方式排列青蛙。具有全局最优适应度的青蛙量子位置为 \mathbf{x}_g 。

Step 03 将全部群体分为 m 个族群。划分规则：第 1 只青蛙进入第 1 个族群，第 2 只青蛙进入第 2 个族群，……，第 m 只青蛙进入第 m 个族群。然后，第 $m+1$ 只青蛙又进入第 1 个族群，依此类推，直到全部青蛙划分完毕。

Step 04 令 $i_m = 0$ ， i_m 为族群序号。

Step 05 令 $i_m = i_m + 1$ ， $i_{gen} = 0$ ， i_{gen} 为族群内迭代次数的序号。族群内最好适应度值和最差适应度值的青蛙量子位置分别记录为 \mathbf{x}_b 和 \mathbf{x}_w 。

Step 06 令 $i_{gen} = i_{gen} + 1$ 。

Step 07 在族群 i_m 内，按照量子蛙跳算法的更新策略更新最差适应度值的青蛙量子位置 \mathbf{x}_w 。首先，按照如下方程更新 \mathbf{x}_w ： $\theta_{wj} = \text{rand} \cdot (\mathbf{x}_{wj} - \mathbf{x}_{bj}) + \text{randn} \cdot (\mathbf{x}_{wj} - \mathbf{z}_j)$ ， $\mathbf{x}_{wj}^{\text{new}} = |\mathbf{x}_{wj} \cos \theta_{wj} - \sqrt{1 - (\mathbf{x}_{wj})^2} \sin \theta_{wj}|$ ， $j = 1, 2, \dots, M$ 。

如果新的青蛙量子位置的适应度值没有改进，则在更新策略中使用 \mathbf{x}_g 代替 \mathbf{x}_b ，使其更快地向全局最优解收敛， $\theta_{wj} = \text{rand} \cdot (\mathbf{x}_{wj} - \mathbf{x}_{gj}) + \text{randn} \cdot (\mathbf{x}_{wj} - \mathbf{z}_j)$ ， $\mathbf{x}_{wj}^{\text{new}} = |\mathbf{x}_{wj} \cos \theta_{wj} - \sqrt{1 - (\mathbf{x}_{wj})^2} \sin \theta_{wj}|$ ， $j = 1, 2, \dots, M$ 。如果适应度值仍然没有改进，

那么在量子域内随机产生新的青蛙量子位置 $\mathbf{x}_w^{\text{new}}$ 。这样便完成了族群 i_m 的一次循环，更新该族群内的 \mathbf{x}_b 、 \mathbf{x}_w 和 \mathbf{x}_g 。

Step 08 如果 $i_{\text{gen}} < \text{gen}$ ，则进入 **Step 06**；否则，执行 **Step 09**。

Step 09 如果 $i_m < m$ ，则进入 **Step 05**。

Step 10 测试终止条件。如果达到最大进化代数，则输出最终 \mathbf{x}_g 及其对应的最优解，终止算法；否则，按照适应度降序的方式排列青蛙，进入 **Step 03**。

3.1.3 性能测试

为了测试量子蛙跳算法的优越性与高效性，选择 4 个通用测试函数^[5,6]对其进行性能测试，并且将测试结果与混合蛙跳算法（SFLA）^[1,3]、粒子群算法（PSO）^[7]及细菌觅食算法（BFOA）^[8]进行了比较。所选取的测试函数为 Sphere 函数、Sum Squares 函数、Griewank 函数和 Ackley 函数，其公式分别为

$$f_1(\mathbf{x}) = \sum_{j=1}^M x_j^2, x_j \in [-5.12, 5.12], (j=1, 2, \dots, M) \quad (3.1.11)$$

$$f_2(\mathbf{x}) = \sum_{j=1}^M jx_j^2, x_j \in [-10, 10], (j=1, 2, \dots, M) \quad (3.1.12)$$

$$f_3(\mathbf{x}) = \frac{1}{4000} \sum_{j=1}^M x_j^2 - \prod_{j=1}^M \cos\left(\frac{x_j}{\sqrt{j}}\right) + 1, x_j \in [-10, 10], (j=1, 2, \dots, M) \quad (3.1.13)$$

$$f_4(\mathbf{x}) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{M} \sum_{j=1}^M x_j^2}\right) - \exp\left(\frac{1}{M} \sum_{j=1}^M \cos 2\pi x_j\right), x_j \in [-15, 30], (j=1, 2, \dots, M) \quad (3.1.14)$$

量子蛙跳算法的参数设置如下：青蛙个体总数 $p=50$ ，族群数 $m=10$ ，每个族群青蛙数 $n=5$ ，族群内最大迭代次数 $\text{gen}=5$ ，最大演进代数 $\text{Gen}=1000$ 。

对于混合蛙跳算法，参数 p 、 m 、 n 、 gen 、 Gen 与量子蛙跳算法保持一致，另外跳跃步长的最大值 d_j^{\max} 和最小值 d_j^{\min} 分别设置为： $d_j^{\max}=0.5$ ， $d_j^{\min}=-0.5$ 。

细菌觅食算法的参数设置为:种群规模 $S=50$, 一次趋向操作中趋向次数 $N_c=100$, 最大连续游弋次数 $N_s=4$, 复制次数 $N_{re}=5$, 迁徙次数 $N_{ed}=2$, 迁徙操作中迁徙概率 $P_{ed}=0.1$, 游弋步长 $c(i)=0.01$ 。粒子群算法的参数设置为:学习因子 $c_1=c_2=2$, 粒子总数为 50, 全部粒子的速度变化范围限定为定义区间的 10%。在整个测试过程中测试函数的维数取 $M=35$, 试验的迭代次数为 1000, 运行 200 次独立试验。

仿真后得到的收敛曲线如图 3.1.1~图 3.1.4 所示。由仿真图可知, 尽管混合蛙跳算法和细菌觅食算法的收敛速度很快, 但是收敛精度不高; 而粒子群算法收敛精度较高, 但是收敛速度较慢。在所有 4 种智能算法中, 量子蛙跳算法无论是收敛速度还是收敛精度都是最优的。

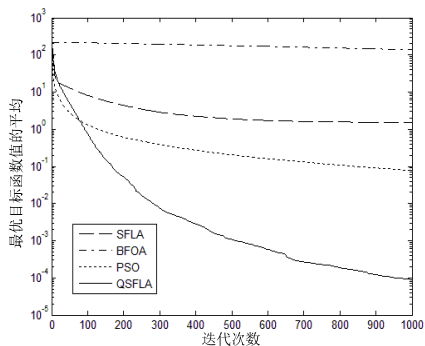


图 3.1.1 Sphere 函数的收敛曲线

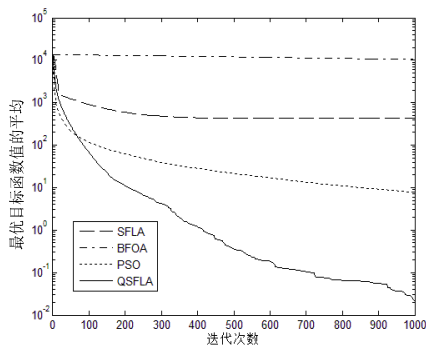


图 3.1.2 Sum Squares 函数的收敛曲线

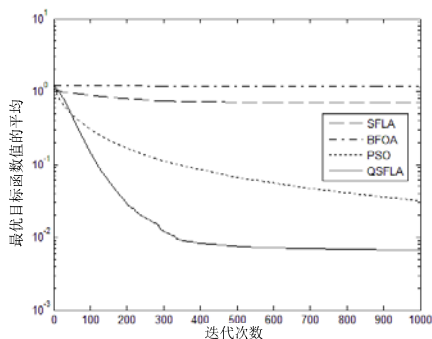


图 3.1.3 Griewank 函数的收敛曲线

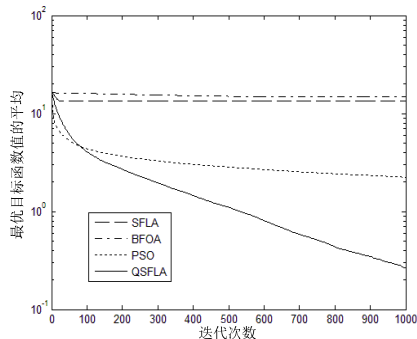


图 3.1.4 Ackley 函数的收敛曲线

3.2 量子文化蛙跳算法^[9]

本节介绍如何把文化算法引入到量子蛙跳算法中，针对最大值优化问题设计了量子规范知识并将其应用在量子蛙跳算法的演进过程当中，可有效求解连续优化问题，仿真测试证明了其可行性和优越性。

3.2.1 量子规范知识

文化是保存信息的载体，它将个体的以往经验保存于知识库，新的个体可以在知识库中学到它没有直接经历的经验知识，因而有效地指导并极大地促进和加快了人类社会的进化发展。文化算法^[10,11]是一种基于种群的多进化过程的计算模型，为进化搜索机制和知识存储的结合提供了一个架构，它最早于 1994 年由 Reynolds 基于文化系统的进化模型提出。文化算法分别从微观和宏观两个层面上模拟文化的双重进化继承过程，从进化角度看，任何一种符合文化算法要求的计算架构都能被用来表示种群空间；不同符号表示方法都能被用来描述信仰空间。任何一种符合文化算法要求的进化算法都可以嵌入文化算法框架中作为种群空间的一个进化过程。

在 M 维优化问题中，算法开始时在可行域内随机创建一个由 p 只青蛙组成的种群。第 k 只量子青蛙的量子位置表示为 $\mathbf{x}_k = [x_{k1}, x_{k2}, \dots, x_{kM}]$ ，而且通过它们的适应性对所有量子青蛙进行降序排列。所有量子青蛙被分成 m 个族群，每个族群都是一个独立演化的群体，每个族群都包含 n 只量子青蛙（即 $p = m \times n$ ）。第 1 只量子青蛙到第 1 个族群，第 2 只量子青蛙到第 2 个族群，……，第 m 只量子青蛙到第 m 个族群，第 $(m+1)$ 只量子青蛙返回到第 1 个族群，依此类推，直到所有的量子青蛙被划分完毕。在每次迭代中，每个族群中最好适应度和最差适应度的量子青蛙的量子位置用 $\mathbf{x}_b = [x_{b1}, x_{b2}, \dots, x_{bM}]$ 和 $\mathbf{x}_w = [x_{w1}, x_{w2}, \dots, x_{wM}]$ 分别记录，而且在所有族群中最好适应度的量子青蛙的量子位置用 $\mathbf{x}_g = [x_{g1}, x_{g2}, \dots, x_{gM}]$ 记录。

接受函数和影响函数作为信仰空间和量子青蛙种群空间之间的通信信道。对于不同的问题,不同特定的知识源在生成最优新量子位置时可能会更加有用。在 QCSFLA 算法中,只考虑两种知识源:量子规范知识和量子形势知识(量子形势知识也就是所有族群中最好适应度的量子青蛙的量子位置)。信仰空间中规范知识为 $\{N_i, i=1,2,\dots,M\}$, 其中 N_i 是包含参数范围信息的量子规范知识。用 $\langle I_i, L_i, U_i \rangle$ 定义 N_i , 其中 I_i 表示第 i 个参量的变化范围, $I_i = [l_i, u_i]$, 下边界 l_i 和上边界 u_i 被初始化为给定的边界值, 而且可以在演变过程中通过接受函数获得个体进行更新。 L_i 代表下边界 l_i 的评价值, U_i 代表上边界 u_i 的评价值。对于极大值优化问题, L_i 和 U_i 被初始化为 $-\infty$ 。

量子规范知识 N_i 将会由以下函数进行更新。假设第 k 只量子青蛙影响第 i 个规范知识的下边界, 下边界和它的性能得分在下面给出。

$$l_i^{h+1} = \begin{cases} x_{ki}^h, & l_i^h \text{ 或 } f(\bar{\mathbf{x}}_k^h) > L_i^h \\ l_i^h, & \text{其他} \end{cases} \quad (3.2.1)$$

$$L_i^{h+1} = \begin{cases} f(\bar{\mathbf{x}}_k^h), & x_{ki}^h, l_i^h \text{ 或 } f(\bar{\mathbf{x}}_k^h) > L_i^h \\ L_i^h, & \text{其他} \end{cases} \quad (3.2.2)$$

其中, l_i^h 表示在第 h 次迭代中第 i 个规范知识的下边界, L_i^h 表示它的评价值。

假设现在第 k 只量子青蛙影响第 i 个规范知识的上边界。

$$u_i^{h+1} = \begin{cases} x_{ki}^h, & u_i^h \text{ 或 } f(\bar{\mathbf{x}}_k^h) > U_i^h \\ u_i^h, & \text{其他} \end{cases} \quad (3.2.3)$$

$$U_i^{h+1} = \begin{cases} f(\bar{\mathbf{x}}_k^h), & x_{ki}^h, u_i^h \text{ 或 } f(\bar{\mathbf{x}}_k^h) > U_i^h \\ U_i^h, & \text{其他} \end{cases} \quad (3.2.4)$$

其中, u_i^h 表示在第 h 次迭代中第 i 个规范知识的上边界, U_i^h 是它的评价值。

3.2.2 量子文化蛙跳算法的实现

假设 $\bar{\mathbf{x}}_k = [\bar{x}_{k1}, \bar{x}_{k2}, \dots, \bar{x}_{kM}]$ 是第 k 只青蛙的位置, 其量子位置映射状态为

$\mathbf{x}_k = [x_{k1}, x_{k2}, \dots, x_{kM}]$, $0 \leq x_{ki} \leq 1$ ($i=1, 2, \dots, M$)。由 $\bar{x}_{ki} = \bar{x}_i^{\min} + x_{ki}(\bar{x}_i^{\max} - \bar{x}_i^{\min})$, $\bar{x}_{ki} \in [\bar{x}_i^{\min}, \bar{x}_i^{\max}]$ 可得第 i 维位置, 其中 \bar{x}_i^{\min} 是第 i 维变量下限, \bar{x}_i^{\max} 是第 i 维变量上限。量子青蛙 k 的适应性由 $f(\bar{\mathbf{x}}_k)$ 计算。

量子青蛙通过使用量子机制和文化机制的演化方式模拟青蛙的跳跃过程来更新它的量子位置和位置。量子青蛙的跳跃方程的更新步骤由以下策略决定。

策略 1: 产生 $[0,1]$ 间的均匀随机数 r_1 , 若其值 $r_1 < p_1$, 则在这个族群中具有最差适应度的量子青蛙的量子旋转角由以下等式进行更新

$$\delta_{wi} = R(0,1) \cdot (x_{wi} - x_{bi}) + G(0,1) \cdot (x_{wi} - z_i) \quad (3.2.5)$$

其中, $i=1, 2, \dots, M$, $R(0,1)$ 是在 $[0,1]$ 内生成均匀分布的随机数, $G(0,1)$ 是一个满足高斯分布、均值为 0、方差为 1 的随机数, δ_{wi} 是 x_{wi} 的量子旋转角, 若 \mathbf{z} 为第 k_m 个族群内全部青蛙的量子位置的平均值

$$\mathbf{z} = [z_1, z_2, \dots, z_M] \quad (3.2.6)$$

其中, $z_i = \frac{1}{n} \sum_{k=1}^n x_{[m(k-1)+k_m], i}$, $i=1, 2, \dots, M$ 。否则, 量子旋转角由以下等式进行更新

$$\delta_{wi} = \begin{cases} -\frac{\pi}{2} |\text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1)|, & x_{wi} < x_{bi} \\ \frac{\pi}{2} |\text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1)|, & x_{wi} > x_{bi} \\ \frac{\pi}{2} \cdot \eta \cdot \text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1), & \text{else} \end{cases} \quad (3.2.7)$$

其中, $i=1, 2, \dots, M$, η 是一个常量 0.06, $\text{size}(\mathbf{I}_i^{h+1}) = u_i^{h+1} - l_i^{h+1}$ 表示信仰空间第 i 个参数的可调节长度。

量子位置 \mathbf{x}_w 由以下等式进行更新

$$x_{wi}^{\text{new}} = |x_{wi} \cos \delta_{wi} - \sqrt{1 - (x_{wi})^2} \sin \delta_{wi}| \quad (3.2.8)$$

如果这次跳跃产生了更好的量子位置 $\mathbf{x}_w^{\text{new}}$, 它将会替代量子位置 \mathbf{x}_w ; 否则, 式 (3.2.5) 和式 (3.2.7) 中用 \mathbf{x}_g 变量替换 \mathbf{x}_b 对应变量执行策略 2 进行计算。

策略 2: 产生 $[0,1]$ 间的均匀随机数 r_2 ，若其值 $r_2 < p_2$ ，则在这个族群中具有最差适应度的量子青蛙的量子旋转角由以下等式进行更新

$$\delta_{wi} = R(0,1) \cdot (x_{wi} - x_{gi}) + G(0,1) \cdot (x_{wi} - z_i) \quad (3.2.9)$$

否则，量子旋转角由以下等式进行更新

$$\delta_{wi} = \begin{cases} -\frac{\pi}{2} |\text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1)|, & x_{wi} < x_{gi} \\ \frac{\pi}{2} |\text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1)|, & x_{wi} > x_{gi} \\ \frac{\pi}{2} \cdot \eta \cdot \text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1), & \text{else} \end{cases} \quad (3.2.10)$$

量子位置 \mathbf{x}_w 由以下等式进行更新

$$\mathbf{x}_{wi}^{\text{new}} = |\mathbf{x}_{wi} \cos \delta_{wi} - \sqrt{1 - (\mathbf{x}_{wi})^2} \sin \delta_{wi}| \quad (3.2.11)$$

如果这次跳跃产生了更好的量子位置 $\mathbf{x}_w^{\text{new}}$ ，它将会替代量子位置 \mathbf{x}_w 。如果在这种情况下没有任何改进，则执行策略 3：产生 $[0,1]$ 间的均匀随机数 r_3 ，若其值 $r_3 < p_3$ ，则在量子位置的定义域内随机产生量子位置作为最差青蛙的新位置；否则，由量子规范知识产生的 $\mathbf{x}_w^{\text{new}}$ 将会替换 \mathbf{x}_w 。

$$\delta_{wi} = \begin{cases} -|\text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1)|, & x_{wi} < l_i^{h+1} \\ |\text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1)|, & x_{wi} > u_i^{h+1} \\ \eta \cdot \text{size}(\mathbf{I}_i^{h+1}) \cdot G(0,1), & \text{else} \end{cases} \quad (3.2.12)$$

$$\mathbf{x}_{wi}^{\text{new}} = |\mathbf{x}_{wi} \cos \delta_{wi} - \sqrt{1 - (\mathbf{x}_{wi})^2} \sin \delta_{wi}| \quad (3.2.13)$$

其中， $i = 1, 2, \dots, M$ 。

在特定迭代次数内重复量子位置的更新操作。在完成全部族群的局部搜索后，种群中所有的量子青蛙进入重新混洗排序过程。

量子文化蛙跳算法的迭代过程叙述如下。

Step 01 参数初始化。赋予青蛙群体规模 p 、青蛙族群数 m 和每个族群内的青蛙数 n 以适当的数值。同样还需要设置族群内的最大迭代次数 gen 及整个群

体的最大迭代次数（混洗次数） Gen 。在定义域内随机产生青蛙的初始位置。初始化规范知识。

Step 02 计算每只青蛙的适应度值。

Step 03 按照适应度值的降序排列青蛙。具有全局最优适应度的青蛙位置为 \mathbf{x}_g 。根据前 q 只优秀青蛙的位置对信仰空间进行更新，按照知识空间更新规则计算量子规范知识。

Step 04 全部青蛙根据混洗规则分为 m 个族群。

Step 05 设 $k_m=0$ ，其中 k_m 为族群的序号。

Step 06 $k_m = k_m + 1$ ， $k_{gen} = 0$ ， k_{gen} 为族群内迭代次数的序号。计算族群 k_m 内青蛙位置的均值 \mathbf{z} 。

Step 07 令 $k_{gen} = k_{gen} + 1$ 。把族群 k_m 内最好适应度值和最差适应度值的青蛙位置分别记作 \mathbf{x}_b 和 \mathbf{x}_w 。

Step 08 在族群 k_m 内，按照量子文化蛙跳算法的更新策略 1 更新最差位置。如果适应度值没有改进，则需要按照更新策略 2 重新更新最差位置。如果适应度值仍没有改进，则需要按照更新策略 3 更新最差位置。更新具有全局最优适应度的青蛙位置为 \mathbf{x}_g 。

Step 09 如果 $k_{gen} < gen$ ，则进入 **Step 07**；否则，执行 **Step 10**。

Step 10 如果 $k_m < m$ ，则进入 **Step 06**；否则，执行 **Step 11**。

Step 11 测试终止条件。如果达到最大混洗次数，则输出最终 \mathbf{x}_g 及其对应的最优适应度值，终止算法；否则，按照适应度值降序排列青蛙，更新量子规范知识，进入 **Step 04**。

3.2.3 性能测试

对于 QCSFLA，族群的数量设置为 $m=20$ ，每个族群中量子青蛙的个数设

置为 $n=5$ 。在混排之前每个族群的迭代次数设置为 5，更新信仰空间优秀青蛙数为 $q=20$ ，最大混排迭代次数设置为 $Gen=1000$ ，选择概率设为 $p_1 = p_2 = p_3 = 0.7$ 。用 200 次的仿真结果的平均值来测试所提出的智能算法的性能。4 种算法个体数都设置为 100，其他 3 种算法参数设置如 3.2.2 节所示。在整个测试过程中测试函数的维数取 $M=60$ ，试验的迭代次数为 1000，运行 100 次独立试验。

对于 Griewank 函数和 Ackley 函数，仿真后得到的收敛曲线如图 3.2.1 和图 3.2.2 所示。在所有 4 种智能算法中，与性能较好的量子蛙跳算法比较，量子文化蛙跳算法无论收敛速度还是收敛精度都是最优的。

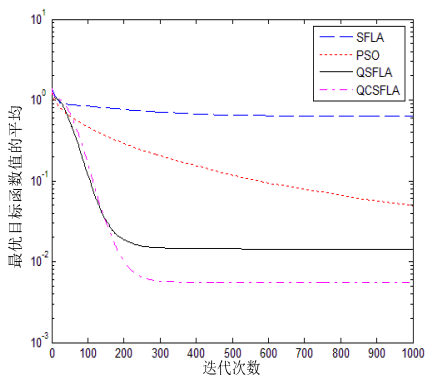


图 3.2.1 Griewank 函数的收敛曲线

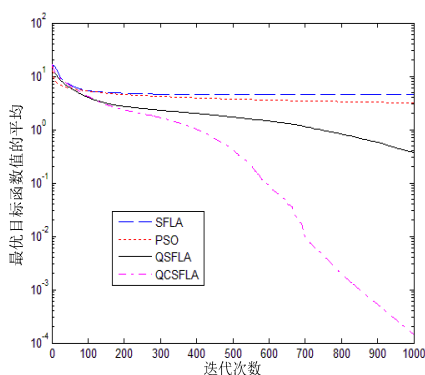


图 3.2.2 Ackley 函数的收敛曲线

3.3 量子细菌觅食算法

随着群体智能算法的蓬勃发展，Passino 于 2002 年提出了模拟人类大肠杆菌觅食行为的细菌觅食优化算法^[12]。细菌觅食算法自提出以来，由于其构造的直观性与易理解的自然机理，吸引了不同领域研究者的密切关注。该算法主要通过趋向性操作、复制操作和迁徙操作这三种操作的迭代来求解问题^[13,14]，但是研究者发现该算法同样具有全局收敛性差的缺陷。趋向性操作是细菌觅食算法最为核心的操作^[15]，本节将量子计算引入趋向性操作之中，设计出全新的量子机制的趋向性操作。通过仿真试验证实新设计的量子细菌觅食算法(QBFOA)具有优异的性能。

假设想寻求函数的最小值，接下来根据细菌觅食算法的基本原理，使用细菌觅食算法和量子细菌觅食算法来解决此优化问题。

3.3.1 细菌觅食算法

1. 趋向性操作

假设细菌种群总数为 S ，每个细菌的位置表示为 $\Theta_i = [\Theta_{i1}, \Theta_{i2}, \dots, \Theta_{iM}]$ ，其中， $i=1, 2, \dots, S$ 。令 $J^i(j, k, l)$ 表示经过第 j 次趋向性操作、第 k 次复制操作、第 l 次迁徙操作之后细菌 i 的适应度值。第 i 个细菌的第 d 维分量在第 j 次趋向性操作时按照如下方式进行

$$\Theta_{id}^{j+1} = \Theta_{id}^j + c(i) \frac{\Delta_d(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (3.3.1)$$

其中， $c(i)$ 表示细菌 i 每次游弋的单位步长， $\Delta(i) = [\Delta_1(i), \dots, \Delta_M(i)]^T$ ，其中每个元素 $\Delta_d(i)$ 是 $[-1, 1]$ 之间的随机数， $d=1, 2, \dots, M$ 。这样经过一次趋向性操作后，细菌 i 的位置便沿着旋转后的方向前进了 $c(i)$ 距离。如果移动后的细菌位置 Θ_i^{j+1} 对应的适应度 $J^i(j+1, k, l)$ 优于 $J^i(j, k, l)$ ，则在该方向上继续前进 $c(i)$ 距离，直到达到事先设定的最大游弋次数 N_s ；如果 $J^i(j+1, k, l)$ 不优于 $J^i(j, k, l)$ ，则结束本次游弋操作，细菌随机旋转寻找下一个移动方向。

2. 复制操作

在进行了 N_c 次趋向性操作后，细菌进行一次复制操作。在复制操作中令 $S_r = \frac{S}{2}$ ，并且定义 $J_{\text{health}}^i = \sum_{j=1}^{N_c+1} J^i(j, k, l)$ 表示细菌 i 的健康值。将细菌种群中的所有细菌按照健康值的降序排列。具有较大健康值的 S_r 个细菌消亡，具有较小健康值的 S_r 个细菌分裂（该过程意味着新复制的子代细菌代替消亡的细菌位于与它们父代细菌相同的位置）。

3. 迁徙操作

在进行了 N_{re} 次复制操作后, 细菌进行一次迁徙操作。对于每一个细菌, 以概率 P_{ed} 进行变异。如果对某一细菌进行变异操作, 则在定义域内随机产生一个新的细菌代替变异细菌。

全体细菌进行 N_{ed} 次迁徙操作之后, 细菌种群便完成了一次进化。

通过上述分析与介绍, 基于量子细菌觅食算法解决优化问题的流程可以归纳如下。

Step 01 初始化参数 $S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, c(i)$ 和 Θ_i , 其中, $i=1, 2, \dots, S$ 。在定义域内随机产生初始细菌位置。

Step 02 迁徙操作循环: $l=l+1$, 令 $k=0$ 。

Step 03 复制操作循环: $k=k+1$, 令 $j=0$ 。

Step 04 趋向性操作循环: $j=j+1$ 。

[a] 对于每个细菌, 按照如下方式进行一次趋向性操作。

[b] 计算适应度函数 $J^i(j, k, l)$ 。

[c] 令 $J_{last}^i = J^i(j, k, l)$ 。

[d] 旋转: 产生一个随机向量 $\Delta(i) \in \mathbf{R}^M$, 其中每个元素 $\Delta_d(i)$ 是 $[-1, 1]$ 之间的随机数, $d=1, 2, \dots, M$ 。

[e] 移动: 第 i 个细菌的第 d 维分量在第 j 次趋向性操作时按照如下方式进行

$$\Theta_{id}^{j+1} = \Theta_{id}^j + c(i) \frac{\Delta_d(i)}{\sqrt{\Delta^T(i) \Delta(i)}}$$

其中, $d=1, 2, \dots, M$, 这样细菌 i 便在该旋转方向上移动了 $c(i)$ 距离。

[f] 计算 $J^i(j+1, k, l)$ 。

[g] 游弋:

(1) 令 $m=0$ 。

(2) while $m < N_s$ 时:

1) 令 $m=m+1$ 。

2) 如果 $J^i(j+1, k, l) < J_{\text{last}}^i$, 则保存 $J_{\text{last}}^i = J^i(j+1, k, l)$, 然后按照如下方式进行

$$\Theta_{id}^{j+1} = \Theta_{id}^{j+1} + c(i) \frac{\Delta_d(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

其中, $d=1, 2, \dots, M$ 。按照[f]步骤, 用 Θ_i^{j+1} 计算 $J^i(j+1, k, l)$ 。

3) 如果 $m=N_s$, 则结束 while 循环。

[h] 如果 $i \neq S$, 则进入[b]步骤处理下一个细菌。

Step 05 如果 $j < N_c$, 则进入 **Step 04** 继续进行趋向性操作。

Step 06 复制操作:

[a] 在给定 k 和 l 的情况下, 对于每一个细菌 $i=1, 2, \dots, S$, 用健康值 $J_{\text{health}}^i = \sum_{j=1}^{N_c+1} J^i(j, k, l)$ 表示第 i 个细菌的健康状况。按照健康值的降序对所有细菌进行排序。

[b] 具有较差 (对于最小值优化问题, 值越大越差) 健康值的 S_r 个细菌消亡, 保留的 S_r 个细菌分裂 (该过程意味着新复制的子代细菌代替消亡的细菌位于与它们父代细菌相同的位置)。

Step 07 如果 $k < N_{re}$, 则进入 **Step 03**, 完成复制操作, 开始进行新一轮的趋向性操作。

Step 08 迁徙操作: 对于每一个细菌, 以概率 P_{ed} 进行变异。如果对某一细菌进行变异, 则在定义域内随机产生一个新的细菌。如果 $l < N_{ed}$, 则进入 **Step 02**; 否则, 终止算法。

3.3.2 量子细菌觅食优化算法^[16]

细菌觅食算法在科学和工程领域已经得到了广泛的应用,为了提高其性能,可根据模拟量子旋转门和改进细菌觅食行为的演化方式设计量子细菌觅食优化算法(QBFOA)。趋向性操作是细菌觅食优化算法的核心操作,在QBFOA中,用量子旋转门来完成趋向性操作。将趋向性操作构建成量子化的随机巡游行为,以便寻求最优解。

在QBFOA中,量子细菌的位置表示为 $\bar{\theta}_i = [\bar{\theta}_{i1}, \bar{\theta}_{i2}, \dots, \bar{\theta}_{iM}]$,其中 $\bar{\theta}_{id} \in [\bar{\theta}_d^{\min}, \bar{\theta}_d^{\max}]$, $d = 1, 2, \dots, M$ 。 $\bar{\theta}_d^{\min}$ 为第 d 维变量的下限, $\bar{\theta}_d^{\max}$ 为第 d 维变量的上限。假定要寻求函数 $J(\bar{\theta}_i)$ 的最小值(其中 $\bar{\theta}_i \in \mathbf{R}^M$), 适应度函数 $J(\bar{\theta}_i)$ 代表外界环境对细菌各种影响的综合度量。根据模拟的量子进化的简化方法,第 i 个量子细菌的量子位置可以表示为 $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iM}]$,第 i 个量子细菌的第 d 维量子位在第 j 次趋向性操作时按照如下方式进行

$$\theta_{id}^{j+1} = |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}| \quad (3.3.2)$$

$\bar{\theta}_i = [\bar{\theta}_{i1}, \bar{\theta}_{i2}, \dots, \bar{\theta}_{iM}]$ 代表量子位置 $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iM}]$ 的映射位置,其中每一量子位置的量子位 $0 \leq \theta_{id} \leq 1$, $d = 1, 2, \dots, M$ 。第 i 个量子细菌的适应度值用 $J(\bar{\theta}_i)$ 来计算。

在一个 M 维优化问题中,在量子细菌觅食算法的开始阶段,在定义区间内随机产生初始量子细菌,初始量子细菌总数为 S 。第 i 个量子细菌的量子位置表示为 $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iM}]$ 。当寻找到最优适应度值时,保存此时量子细菌的量子位置 $\mathbf{b} = [b_1, b_2, \dots, b_M]$ 。按照如下趋向性操作来更新量子位置

$$\phi_{id}^{j+1} = r_1 \cdot (\theta_{id}^j - b_d^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j) \quad (3.3.3)$$

$$\theta_{id}^{j+1} = |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}| \quad (3.3.4)$$

其中, r_1 和 r_2 是区间 $[0,1]$ 内的均匀随机数, r 是区间 $[-1,1]$ 内的均匀随机数。 $\text{sign}(r)$ 表示取符号函数,其值在 $\{-1, 0, 1\}$ 之中。 β 是一个权重因子, z^j 为全部 S 个量子细菌的量子位置的平均值。

$$\mathbf{z}^j = [z_1^j, z_2^j, \dots, z_M^j] \quad (3.3.5)$$

基于上述分析与介绍，量子细菌觅食算法的流程可以归纳如下。

Step 01 初始化参数 S 、 N_c 、 N_s 、 N_{re} 、 N_{ed} 、 P_{ed} 和 $\theta_i, i=1,2,\dots,S$ 。初始化的量子位置在量子域内随机产生。

Step 02 迁徙操作循环： $l = l+1, k=0$ 。

Step 03 复制操作循环： $k = k+1, j=0$ 。

Step 04 趋向性操作循环： $j = j+1$ 。

[a] 保存当前量子细菌的量子位置 $\theta_i^j, i=1,2,\dots,S$ ，令 $i=1$ 。

[b] 把量子位置映射到定义区间的位置，计算适应度，寻找当前的最优适应度值，保存相关量子细菌的最优量子位置 $\mathbf{b}^j = [b_1^j, b_2^j, \dots, b_M^j]$ 。

[c] 对于每个量子细菌，按照如下方式进行一次趋向性操作。

[d] 当前细菌 i 的适应度值为 $J^i(j, k, l)$ 。

[e] 令 $J_{last}^i = J^i(j, k, l)$ 。

[f] 计算全部 S 个量子细菌的量子位置的平均值 $\mathbf{z}^j = [z_1^j, z_2^j, \dots, z_M^j]$ 。

[g] 旋转和移动：根据式 (3.3.3) 和式 (3.3.4) 计算 $\phi_{id}^{j+1} = r_1 \cdot (\theta_{id}^j - b_d^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j)$ ， $\theta_{id}^{j+1} = |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}|$ ，其中， $d=1,2,\dots,M$ 。

[h] 用映射位置 $\bar{\theta}_i^{j+1} = [\bar{\theta}_{i1}^{j+1}, \bar{\theta}_{i2}^{j+1}, \dots, \bar{\theta}_{iM}^{j+1}]$ 计算适应度值 $J^i(j+1, k, l)$ 。

[i] 游弋：

1) 令 $m=0$ 。

2) while $m < N_s$ 时：

• 令 $m=m+1$ 。

• 如果 $J^i(j+1, k, l) < J_{last}^i$ ，则保存 $J_{last}^i = J^i(j+1, k, l)$ ， $\theta_i^j = \theta_i^{j+1}$ ，即量子细

菌把游弋操作的最优位置放在 θ_i^j 中，根据式 (3.3.3) 和式 (3.3.4) 计算 $\phi_{id}^{j+1} = r_1 \cdot (\theta_{id}^j - b_d^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j)$ ， $\theta_{id}^{j+1} = |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}|$ ，其中 $d=1,2,\dots,M$ 。按照 [h] 步骤，用新的映射位置 $\bar{\theta}_i^{j+1} = [\bar{\theta}_{i1}^{j+1}, \bar{\theta}_{i2}^{j+1}, \dots, \bar{\theta}_{iM}^{j+1}]$ 计算 $J^i(j+1, k, l)$ 。

- 否则，如果 $m = N_s$ ，则结束 while 循环，设置 $\theta_i^{j+1} = \theta_i^j$ 。

[j] 如果 $i \neq S$ ，则进入 [b] 步骤处理下一个量子细菌。

Step 05 如果 $j < N_c$ ，则进入 **Step 04** 继续进行趋向性操作。

Step 06 复制操作：

[a] 在给定 k 和 l 的情况下，对于每一个量子细菌 $i=1,2,\dots,S$ ，使得健康值 $J_{\text{health}}^i = J^i(N_c, k, l)$ 表示第 i 个量子细菌的健康状况。按照健康值的降序对所有量子细菌进行排序。

[b] 具有较差健康值的 S_r 个量子细菌消亡，保留的 S_r 个量子细菌分裂（该过程意味着新复制的子代量子细菌代替消亡的量子细菌位于与它们父代量子细菌相同的位置）。

Step 07 如果 $k < N_{re}$ ，则进入 **Step 03**。完成复制操作，开始进行新一轮的趋向性操作。

Step 08 迁徙操作：对于每一个非最优量子细菌，以概率 P_{ed} 进行变异。如果对某一量子细菌进行变异，则在量子域内随机产生一个新的量子细菌。如果 $l < N_{ed}$ ，则进入 **Step 02**；否则，终止算法。

为了计算与比较的需要，令 ite 表示 BFOA 和 QBFOA 算法的当前迭代次数， $ite = N_c \cdot (k-1) + N_c \cdot N_{re} \cdot (l-1) + j$ 。

3.3.3 性能测试

为了验证量子细菌觅食算法的优越性与高效性，选择 4 个通用测试函数对

其进行了测试，并且将测试结果与混合蛙跳算法^[1,2]、粒子群算法^[7]及细菌觅食算法^[8]进行了比较。所选取的测试函数为 Ackley 函数、Griewank 函数、Sphere 函数和 Rosenbrock 函数，其公式分别为

$$f_1(\bar{\theta}) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{M} \sum_{d=1}^M \bar{\theta}_d^2}) - \exp(\frac{1}{M} \sum_{d=1}^M \cos 2\pi \bar{\theta}_d) \quad (3.3.6)$$

其中， $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_M]$ 并且 $\bar{\theta}_d \in [-15, 15], d = 1, \dots, M$ 。

$$f_2(\bar{\theta}) = \frac{1}{4000} \sum_{d=1}^M \bar{\theta}_d^2 - \prod_{d=1}^M \cos(\frac{\bar{\theta}_d}{\sqrt{d}}) + 1 \quad (3.3.7)$$

其中， $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_M]$ 并且 $\bar{\theta}_d \in [-10, 10], d = 1, \dots, M$ 。

$$f_3(\bar{\theta}) = \sum_{d=1}^M \bar{\theta}_d^2 \quad (3.3.8)$$

其中， $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_M]$ 并且 $\bar{\theta}_d \in [-5.12, 5.12], d = 1, \dots, M$ 。

$$f_4(\bar{\theta}) = \sum_{d=1}^{M-1} (100(\bar{\theta}_{d+1} - \bar{\theta}_d)^2 + (\bar{\theta}_d - 1)^2) \quad (3.3.9)$$

其中， $\bar{\theta} = [\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_M]$ 并且 $\bar{\theta}_d \in [-2, 2], d = 1, \dots, M$ 。

量子细菌觅食算法的主要参数设置如下： $S=50$ ， $N_c=100$ ， $N_s=2$ ， $N_{re}=5$ ， $N_{ed}=2$ ， $P_{ed}=0.1$ ， $\beta=1$ 。对于细菌觅食算法，参数与量子细菌觅食算法保持一致，另外单位游弋步长设置为 $c(i)=0.01$ 。

混合蛙跳算法的参数设置为：青蛙群体的个数总数 $p=50$ ，族群数 $m=10$ ，每个族群青蛙数 $n=5$ ，族群内最大迭代次数 $gen=5$ ，最大演进代数 $Gen=1000$ 。另外 d_j^{\max} 和 d_j^{\min} 设置为： $d_j^{\max}=0.5$ ， $d_j^{\min}=-0.5$ 。细菌觅食算法的参数设置为： $S=50$ ， $N_c=100$ ， $N_s=4$ ， $N_{re}=5$ ， $N_{ed}=2$ ， $P_{ed}=0.1$ ， $c(i)=0.01$ 。粒子群算法的参数设置为：学习因子 $c_1=c_2=2$ ，粒子总数为 50，全部粒子的速度变化范围限定为定义区间的 10%。

在整个测试过程中测试函数的维数取 $M=40$ ，试验的迭代次数为 1000，运行 200 次独立试验。

仿真后得到的收敛曲线如图 3.3.1~图 3.3.4 所示。通过仿真图的结果可知，在所有 4 种智能算法中，量子细菌觅食算法无论收敛速度还是收敛精度都是最优的。

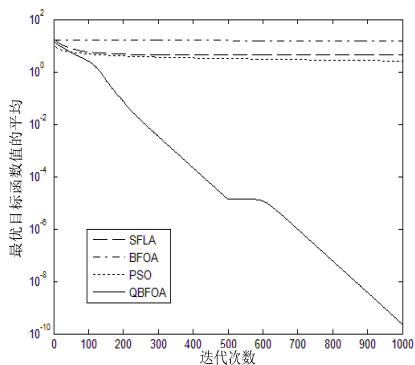


图 3.3.1 Ackley 函数的收敛曲线

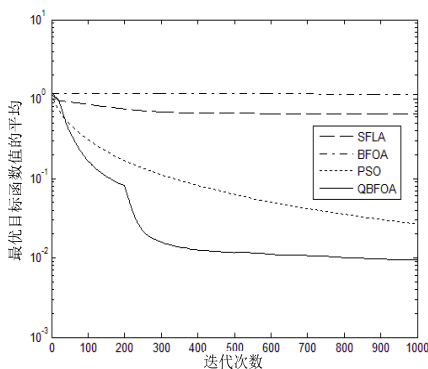


图 3.3.2 Griewank 函数的收敛曲线

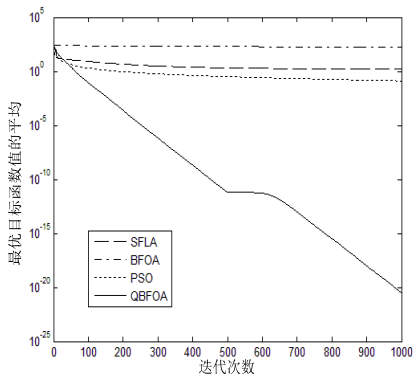


图 3.3.3 Sphere 函数的收敛曲线

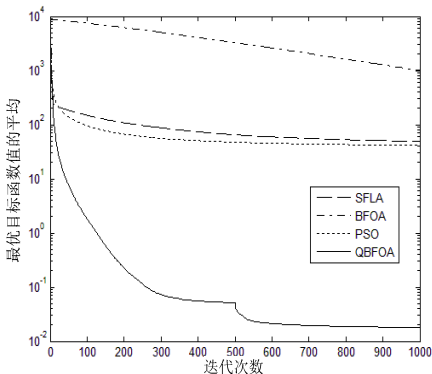


图 3.3.4 Rosenbrock 函数的收敛曲线

3.4 小结

本章主要介绍了创新性设计的量子群智能算法——量子蛙跳算法和量子细菌觅食算法，并且将它们与传统蛙跳算法、细菌觅食算法和粒子群算法等典型

演化算法进行了比较和仿真。由 Benchmark 函数测试的仿真结果可知,量子群智能算法与经典群智能算法相比,具有更快的收敛速度和更高的收敛精度,从而验证了所提出的算法的有效性和优越性。

参考文献

- [1] Eusu M M, and Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.
- [2] Zhen Z, Wang D, Liu Y. Improved shuffled frog leaping algorithm for continuous optimization problem. IEEE Congress on Evolutionary Computation, 2009:2992-2995.
- [3] Elbeltagi E, Hegazy T, Grierson D. Comparison among five evolutionary-based optimization algorithms. Advanced Engineering Informatics. 2005, 19(1): 43-53.
- [4] Hongyuan Gao, Wen Cui , Yilin Jiang. A quantum.-inspired shuffled frog leaping algorithm and its application in cognitive radio. International Journal of Digital Content Technology and its Applications.2012, 20(6):32-42.
- [5] Whitley D, Beveridge R, Graves C, Mathias K. Test driving three genetic algorithms: new test functions and geometric matching. Journal of Heuristics. 1995, 1(1):77-104.
- [6] 王凌. 智能优化算法及其应用. 北京: 清华大学出版社, 2001.
- [7] Kennedy J. Particle swarm optimization. IEEE International Conference on Neural Networks, 1995, 4:1942-1948.
- [8] Passino K M. Biomimicry of bacterial foraging for distributed optimization

- p and control. IEEE Control Systems Magazine. 2002, 22(3):52-67.
- [9] Hongyuan Gao, Congqiang Xu. Cultural quantum-inspired shuffled frog leaping algorithm for direction finding of non-circular signals. International Journal of Computing Science and Mathematics.2013,4(4):321-331.
 - [10] Reynolds R G, Zhu S. Knowledge-based function optimization using fuzzy cultural algorithms with evolutionary programming. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. 2001, 31(1):1-17.
 - [11] Hongyuan Gao and Ming Diao. Cultural firework algorithm and its application for digital filters design. International Journal of Modelling, Identification and Control. 2011, 14 (4):324 - 331
 - [12] Liu Y, Passino K.M. Biomimicry of social foraging bacteria for distributed optimization models, principles, and emergent behaviors. Journal of Optimization Theory and Applications. 2002, 115(3): 603-628.
 - [13] Rashtchi V, Bayat A, Vahedi H. Adaptive step length bacterial foraging algorithm. IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009, 1:20-22.
 - [14] Das S, Biswas A, Dasgupta S, Abraham A. Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. Foundations of Computational Intelligence.2009, 3:23-55.
 - [15] Das S, Dasgupta S, Dasgupta S, Biswas A, Abraham A, Konar A. On stability of the chemotactic dynamics in bacterial foraging optimization algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans. 2009, 39(3): 670-679.
 - [16] Hongyuan Gao, Wen Cui, Chenwan Li. A quantum bacterial foraging optimisation algorithm and its application in spectrum sensing. International Journal of Modelling, Identification and Control. 2013, 18(3): 234-242.

第 4 章

基于量子智能算法的多用户检测

CDMA 系统虽然有很多优点，但它也存在一些问题。最主要的问题是由于不同用户之间的扩频信号不完全正交，会产生多址干扰（Multiple Access Interference, MAI）。由于多址干扰的存在，经常会发生近地强信号压制远地弱信号的现象，这种现象被称为远近效应。这一问题并不是 CDMA 系统固有的问题，而是将匹配滤波器这一单用户高斯信道的最优检测器应用于存在多用户干扰系统中的必然结果，因此抗多址干扰和远近效应技术成为 CDMA 系统的重要课题。

抑制多址干扰的传统方法可通过如下手段实现：设计正交性优异的扩频序列；采用严格功率控制的方法；利用前向纠错 FEC 技术；利用天线空间分集、小区扇区化及波束形成；利用 Rake 接收机等。但是这几种多址干扰抑制方法都没有考虑多址干扰存在的结构特征，因此只能在一定程度上抑制多址干扰，却不能从根本上解决 CDMA 系统中的多址干扰和远近效应问题。

如果从信息论的角度来看多址干扰，它与高斯白噪声有着本质的不同，它携带着正在通信的用户信息，是可以为接收者提供一些先验知识的信号，如其他用户的扩频波形信息，若将这些信息丢掉或当成噪声，必然会造成整个 CDMA 系统容量的下降和恶化 CDMA 系统的误码率性能。多用户检测技术就

是研究如何利用信息论,进而通过严格的理论分析后提出的一种新型抗多址干扰和远近效应的技术。随着研究的深入,通过设计多用户检测接收设备不仅可以实现抗多址干扰和远近效应,还可以抵抗多径干扰和衰落,进而大幅度提高系统容量。由此可见,对多用户检测技术的深入研究具有重要的理论价值和现实意义。

当前,基于智能计算的多用户检测技术在工程应用中的主要障碍是收敛性能和收敛速度之间的矛盾。以解决优化问题的遗传算法为例,在有较大长度待优化数据时,为了避免早熟收敛,要求有大的种群规模和迭代次数,这在实时实现上要付出很大的代价。因此,深入研究要解决的问题,设计实时性好、检测性能高的智能多用户检测算法是一个值得努力的方向。

由于多用户信号的最佳检测可以归结为在数学上求解极大似然函数的最大值,是典型的组合优化问题,所以近年来在 CDMA 系统中采用智能算法的智能信号处理技术去解决多用户检测问题引起了人们的兴趣。初步研究表明,所有解决组合优化问题的二进制编码的智能计算方法都能用于多用户检测问题的求解。

遗传算法、进化规划、粒子群优化、蚁群算法、克隆选择算法、禁忌搜索、模拟退火和遗传量子算法等算法都可以用于多用户检测^[1-6],但大量的仿真实践表明,要想获得好的仿真结果,单独使用智能优化算法无法在有限的时间内获得全局最优解。

由于各优化算法本身的局限性,虽然很多优化算法具有较好的通用性,但在多用户检测这个工程问题上,单独使用一种智能算法收敛性能和收敛速度的折中并不能达到一个满意的效果。单一的优化算法在解决多用户问题时遇到的一个难题是算法收敛性能差,即使使用了足够多的计算量,其收敛效果仍不能达到全局最优。一些算法收敛性能好,但搜索过程的随机性较强,收敛速度较慢,需使用足够多的计算量才能近似达到全局最优,依旧解决不了实时性问题。

如何用最少的时间得到多用户检测的最优解是一个具有挑战意义和研究价值的课题。本章针对多用户检测问题自身的特点和优势,使用所设计的量子神

经网络、免疫克隆量子算法和量子蜂群优化等智能计算原理设计多用户检测器，为多用户检测技术的工程应用和理论研究提供一些可供选择的新方法。

4.1 多用户检测的数学模型

4.1.1 噪声模型

在多用户检测中讨论的噪声大多为高斯白噪声，均值为 μ 、方差为 σ^2 的高斯白噪声模型的概率密度函数为

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \quad (4.1.1)$$

而噪声信道中的噪声模型与实际情况有不相符合的情况，一种常用的非高斯噪声模型为二项高斯混合模型，该噪声模型的概率密度函数为

$$f = (1 - \varepsilon)N(0, v^2) + \varepsilon N(0, kv^2) \quad (4.1.2)$$

式中， $v > 0$ ， $0 < \varepsilon < 1$ ， $k > 0$ ， $N(0, v^2)$ 为背景噪声分量， $N(0, kv^2)$ 为脉冲噪声分量， ε 为脉冲噪声出现的概率。近年来，描述冲击噪声的稳定噪声模型作为广泛意义上的噪声模型被深入研究，其对称稳定分布可用特征函数定义如下

$$\phi(t; \alpha, \gamma, \theta) = E\{\exp(jtX)\} = \exp(j\theta t - \gamma|t|^\alpha) \quad (4.1.3)$$

式中， $0 < \alpha \leq 2$ ， $\gamma > 0$ ， $-\infty < \theta < +\infty$ ，因此用三个参量 α 、 γ 和 θ 就可完备地表示对称分布随机变量。 α 为特征指数，表示拖尾的严重程度，其值越小，拖尾越严重。当其值为2时，对应于高斯分布；当其值为1时，对应于柯西分布。 γ 为散射参数，或称离差，在高斯分布时 $\gamma = \frac{1}{2} \text{Var}(X)$ 。 θ 为位置参数，当 $1 < \alpha \leq 2$ 时，它代表均值；当 $0 < \alpha < 1$ 时，它代表中值。通过对特征函数求傅里叶变换，得到对称稳定随机变量 X 的概率密度函数为

$$f(t; \alpha, \gamma, \theta) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \phi(t; \alpha, \gamma, \theta) \exp(-j\omega t) d\omega \quad (4.1.4)$$

4.1.2 高斯噪声下的多用户检测数学模型

由于 CDMA 系统扩频码不完全正交, 传统的匹配滤波接收机不可避免地存在多址干扰和远近效应问题^[7]。多用户检测技术在接收端设计干扰抑制方法, 从收到干扰的接收信息中可靠地解调出目标用户的信号。本节将在异步 CDMA 系统推导多用户检测的数学模型, 同步 CDMA 系统的多用户检测数学模型可看作异步系统的一个特例。

对于直接序列 (Direct Sequence, DS) CDMA, 每个用户都对应一个独有的特征波形, 接收信号是不同特征波形对用户传递信号扩频后在接收端的迭加再加和信道噪声的混合, 如图 4.1.1 所示, 这就是基本的 DS-CDMA 系统基带接收信号模型, 其数学模型可以表示为

$$r(t) = \sum_{i=-M}^M \sum_{k=1}^K b_k^{(i)} A_k s_k(t - iT_b - \tau_k) + n(t) \quad (4.1.5)$$

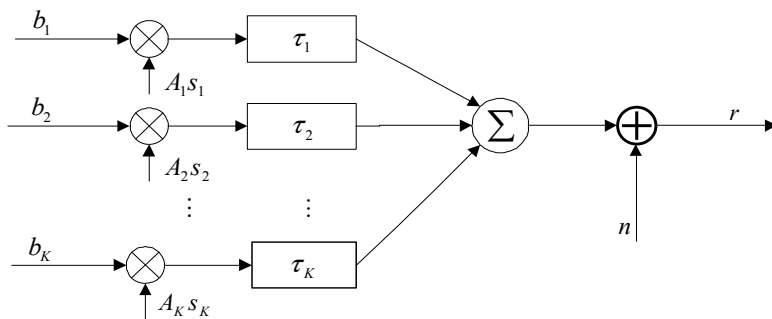


图 4.1.1 DS-CDMA 系统接收信号数学模型

其中, A_k 是用户 k 的信号幅度, $b_k^{(i)} \in \{-1, +1\}$ 表示第 k 个用户传递的第 i 个码元, $s_k(t)$ 是用户 k 的扩频波形, T_b 是每个码元的持续时间, $\tau_k \in [0, T_b)$ 表示用户 k 的时间延迟, 每个用户发射一帧码元的长度等于 $2M + 1$, $n(t)$ 为具有均值为 0 的

双边功率谱密度为 $N_0/2$ 的加性高斯白噪声。第 k 个用户的扩频信号 s_k 可表示为

$$s_k(t) = \sum_{l=0}^{L-1} a_l^{(k)} P_{T_c}(t - lT_c) \quad (4.1.6)$$

P_{T_c} 是持续时间为 T_c 的矩形脉冲, T_c 是码片间隔, $a_l^{(k)}$ 是用户 k 的第 l 个伪随机扩频码片, $L = T_b / T_c$ 为扩频增益。

为了分析和研究的方便, 各个用户的扩频信号要进行归一化处理, 即在一个码周期内有

$$\|s_k\|^2 = \int_0^{T_b} s_k(t) s_k(t) dt = 1 \quad (4.1.7)$$

多用户检测器一般都要使用一个前处理装置, 通过它可以对接收到的连续时间波形 $r(t)$ 进行采样得到离散时间信号 $y_k^{(i)}$ 。最常使用的前处理装置就是使接收信号通过一个匹配滤波器组^[8]。

在异步 CDMA 系统中, 设定每个用户的时间延迟不同, 并且随着用户标号增加延时也增加, 可以给出异步 CDMA 系统的时间偏移为 $0 \quad \tau_1 \quad \tau_2 \quad \dots \quad \tau_K < T_b$ 。

将连续时间波形 $r(t)$ 通过图 4.1.2 所示的匹配滤波器组可以得到一个码元持续时间 T_b 的离散时间信号 $y_k^{(i)}$ 可以写作

$$y_k^{(i)} = \int_{\tau_k + iT_b}^{\tau_k + (i+1)T_b} r(t) s_k(t - iT_b - \tau_k) dt \quad (4.1.8)$$

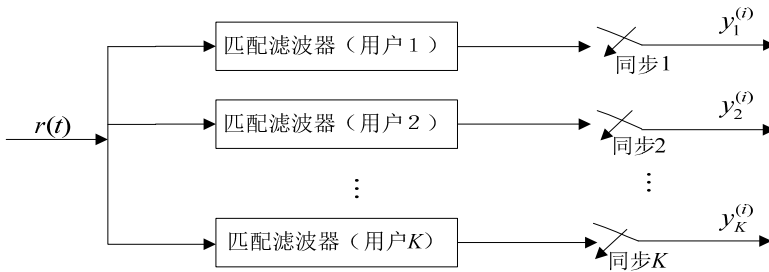


图 4.1.2 匹配滤波器组

将接收信号的表达式 (4.1.5) 代入式 (4.1.8) 中, 得到第 k 个匹配滤波器输出的第 i 个传递码元的离散时间接收数据 $y_k^{(i)}$, 它可表示为

$$y_k^{(i)} = A_k b_k^{(i)} + \sum_{j < k} A_j b_j^{(i+1)} \rho_{kj}(-1) + \sum_{j < k} A_j b_j^{(i)} \rho_{kj}(0) + \sum_{j > k} A_j b_j^{(i-1)} \rho_{kj}(+1) + \sum_{j > k} A_j b_j^{(i)} \rho_{kj}(0) + n_k^{(i)} \quad (4.1.9)$$

$$n_k^{(i)} = \int_{\tau_k + iT_b}^{\tau_k + (i+1)T_b} n(t) s_k(t - iT_b - \tau_k) dt \quad (4.1.10)$$

其中, $\rho_{kj}(-1)$ 、 $\rho_{kj}(0)$ 、 $\rho_{kj}(+1)$ 被用来表示扩频码元间互相关系数。

$$\rho_{kj}(-1) = \int_{\tau_k}^{T_b + \tau_k} s_k(t - \tau_k) s_j(t - T_b - \tau_j) dt \quad (4.1.11)$$

$$\rho_{kj}(0) = \int_{\tau_k}^{T_b + \tau_k} s_k(t - \tau_k) s_j(t - \tau_j) dt \quad (4.1.12)$$

$$\rho_{kj}(+1) = \int_{\tau_k}^{T_b + \tau_k} s_k(t - \tau_k) s_j(t + T_b - \tau_j) dt \quad (4.1.13)$$

进一步将其表示成矩阵向量形式为

$$\mathbf{y}^{(i)} = \mathbf{R}(+1) \mathbf{A}^{(i-1)} \mathbf{b}^{(i-1)} + \mathbf{R}(0) \mathbf{A}^{(i)} \mathbf{b}^{(i)} + \mathbf{R}(-1) \mathbf{A}^{(i+1)} \mathbf{b}^{(i+1)} + \mathbf{n}^{(i)} \quad (4.1.14)$$

其中,

$$\mathbf{y}^{(i)} = [y_1^{(i)}, y_2^{(i)}, \dots, y_K^{(i)}]^T,$$

$$\mathbf{A}^{(i-1)} = \mathbf{A}^{(i)} = \mathbf{A}^{(i+1)} = \text{diag}(A_1, A_2, \dots, A_K),$$

$$\mathbf{b}^{(i)} = [b_1^{(i)}, b_2^{(i)}, \dots, b_K^{(i)}]^T,$$

$$\mathbf{n}^{(i)} = [n_1^{(i)}, n_2^{(i)}, \dots, n_K^{(i)}]^T.$$

其中, $\mathbf{n}^{(i)}$ 为均值是 0、方差是 σ^2 的高斯过程, 其自相关矩阵可表示为

$$E\{\mathbf{n}^{(i)} \mathbf{n}^{(j)T}\} = \begin{cases} \sigma^2 \mathbf{R}(-1), & j = i + 1 \\ \sigma^2 \mathbf{R}(0), & j = i \\ \sigma^2 \mathbf{R}(+1), & j = i - 1 \\ 0, & \text{其他} \end{cases} \quad (4.1.15)$$

由于在异步情况下前后码元间有相互干扰, 因此在进行处理时往往需要将

所有用户在一帧内的离散信号进行联合处理，才可以充分利用其相关信息。此时接收到的数据向量的矩阵形式为

$$\mathbf{y} = \mathbf{R}\mathbf{A}\mathbf{b} + \mathbf{n} \quad (4.1.16)$$

接收的一帧数据为

$$\mathbf{y} = [y_1^{(-M)}, y_2^{(-M)}, \dots, y_K^{(-M)}, \dots, y_1^{(0)}, y_2^{(0)}, \dots, y_K^{(0)}, \dots, y_1^{(M)}, y_2^{(M)}, \dots, y_K^{(M)}]^T,$$

$$\mathbf{A} = \text{diag}(\mathbf{A}^{(-M)}, \dots, \mathbf{A}^{(0)}, \dots, \mathbf{A}^{(M)}),$$

$$\mathbf{b} = [b_1^{(-M)}, b_2^{(-M)}, \dots, b_K^{(-M)}, \dots, b_1^{(0)}, b_2^{(0)}, \dots, b_K^{(0)}, \dots, b_1^{(M)}, b_2^{(M)}, \dots, b_K^{(M)}]^T,$$

$$\mathbf{n} = [n_1^{(-M)}, n_2^{(-M)}, \dots, n_K^{(-M)}, \dots, n_1^{(0)}, n_2^{(0)}, \dots, n_K^{(0)}, \dots, n_1^{(M)}, n_2^{(M)}, \dots, n_K^{(M)}]^T,$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}(0) & \mathbf{R}(-1) & 0 & \dots & 0 & 0 \\ \mathbf{R}(+1) & \mathbf{R}(0) & \mathbf{R}(-1) & \dots & 0 & 0 \\ 0 & \mathbf{R}(+1) & \mathbf{R}(0) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \mathbf{R}(+1) & \mathbf{R}(0) \end{bmatrix} \in \mathbf{R}^{(2M+1)K \times (2M+1)K} \quad (4.1.17)$$

假设所有用户的时间延迟相等，即 $\tau_1 = \tau_2 = \dots = \tau_K = 0$ ，则异步系统就简化为同步系统。同步系统的前后码元之间没有相互影响，因此每个用户只需一个码元就可以分析出信号特性，即取 $M = 0$ 即可。无论是同步还是异步通信系统，接收到的向量 \mathbf{y} 包含了原发送信号的所有信息，称其为解调发送信号 \mathbf{b} 的充分统计量。不同的多用户检测理论就是设计处理这些充分统计量的方法，以达到在某种代价函数最小化的情况下解调出发送信号 \mathbf{b} 。许多准最优多用户检测算法就是对式(4.1.16)表示的离散时间信号进行处理以获得用户传输码元的估计。

同步和异步多用户检测的宏观数学模型是没有区别的，只是异步情况下的计算复杂度更大一些。一般而言，具有 K 个用户的异步 CDMA 系统模型可以改写为有 $2K - 1$ 个用户的等价同步系统。因此，为了分析问题的方便，本章在对所设计的智能多用户检测器进行仿真验证的时候，大多数是在同步 CDMA 系统中进行的，因为对同步系统所采用的方法和所获得的结果都适用于异步系统。

4.2 高斯噪声环境的典型多用户检测方法

为了便于和基于量子智能算法的多用户检测器进行性能比较, 本节介绍三种经典多用户检测器。传统检测器最简单且易于实现, 但性能最差, 抗干扰能力不强; 最优多用户检测器性能最好, 但是计算量与用户数呈指数关系, 最难实现; 而 Hopfield 神经网络多用户检测器是可以快速实现的准最优多用户检测器, 在多址干扰和远近效应较小时效果优良, 但在干扰较大时性能恶化严重。

4.2.1 传统检测器

传统检测器 (Conventional Detector, CD) 的检测原理如图 4.2.1 所示。传统检测器是将接收信号通过 K 个匹配滤波器组成的匹配滤波器组进行相干处理, 先获得对应于 K 个用户的接收数据 $\mathbf{y}^{(i)} = [y_1^{(i)}, y_2^{(i)}, \dots, y_K^{(i)}]^T$, 写成矩阵向量形式为

$$\mathbf{y}^{(i)} = \mathbf{R}(+1)\mathbf{A}^{(i-1)}\mathbf{b}^{(i-1)} + \mathbf{R}(0)\mathbf{A}^{(i)}\mathbf{b}^{(i)} + \mathbf{R}(-1)\mathbf{A}^{(i+1)}\mathbf{b}^{(i+1)} + \mathbf{n} \quad (4.2.1)$$

或者简化写作

$$\mathbf{y} = \mathbf{R}\mathbf{A}\mathbf{b} + \mathbf{n} \quad (4.2.2)$$

然后通过取符号操作对接收数据进行二值硬判决, 得到传统检测器的检测结果为

$$\mathbf{b}_{\text{CD}}^{(i)} = \text{sgn}(\mathbf{y}^{(i)}) \quad (4.2.3)$$

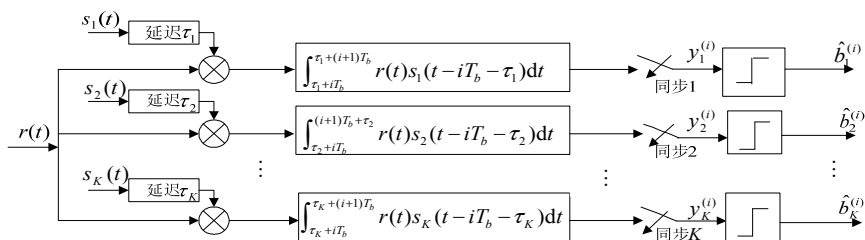


图 4.2.1 传统检测器

同步 CDMA 系统中的传统检测器误码率可以使用公式定量表达为

$$P_k = 2^{1-K} \sum_{\substack{b \in \{-1,1\} \\ b_k = -1}} Q \left[\frac{A_k - \sum_{j \neq k} A_j \rho_{kj} b_j}{\sigma} \right] \quad (4.2.4)$$

式中, A_k 是用户 k 的接收信号幅值, ρ_{kj} 是互相关矩阵 \mathbf{R} 的一个元素, σ^2 是噪声方差。传统检测器的渐近有效性和抗远近能力可分别表示为

$$\eta_k = \max^2 \left\{ 0, 1 - \sum_{j \neq k} \frac{A_j^2}{A_k^2} |\rho_{kj}| \right\} \quad (4.2.5)$$

$$\bar{\eta}_k = \begin{cases} 1 & \rho_{kj} = 0, j \neq k \\ 0 & \text{otherwise} \end{cases} \quad (4.2.6)$$

从以上各式可以看出, 只要扩频序列没有完全正交, 互相关系数就是非 0 的数, 多址干扰和远近效应就不可能避免。

4.2.2 最优多用户检测器

最优多用户检测器采用的是贝叶斯后验概率最大原理, 因为是一种最大似然序列估计算法, 可以达到理论上的最小错误概率^[9]。具体来说, 就是在解空间找到发送信号向量 \mathbf{b} 使得

$$f[\{r(t), t \in [-MT_b, MT_b + 2T_b]\} | \mathbf{b}] = \exp \left[-\frac{1}{2\sigma^2} \int_{-MT_b}^{MT_b + 2T_b} [r(t) - S(t, \mathbf{b})]^2 dt \right] \quad (4.2.7)$$

这个似然函数最大化, 式中

$$S(t, \mathbf{b}) = \sum_{i=-M}^M \sum_{k=1}^K A_k b_k(i) s_k(t - iT_b - \tau_k) \quad (4.2.8)$$

令 \mathbf{A} 为 $K(2M+1) \times K(2M+1)$ 维的对角矩阵, 其中第 $k+iK$ 个对角元素等于 A_k 。记 $S_t(\mathbf{b}) = S(t, \mathbf{b})$, $v_{k+iK}(t) = s_k(t - iT_b - \tau_k)$, 则 $\rho_{kj}(\mathbf{v})$ 是矩阵 $\mathbf{R}(\mathbf{v})$ 的元素, $\rho_{kj}(\mathbf{v})$ 的数学表达由式 (4.1.11) ~ (4.1.13) 给出, $\mathbf{R}(\mathbf{v}) \in \mathbf{R}^{K \times K}$, $\mathbf{v} \in \{-1, 0, +1\}$ 。令 \mathbf{R} 为由式 (4.1.20) 表示的 $K(2M+1) \times K(2M+1)$ 矩阵。在同步的情形下, 有

$M=0$, $\tau_k=0$, $\mathbf{R}=\mathbf{R}(0)$ 。若定义

$$\mathbf{H} = \mathbf{A}\mathbf{R}\mathbf{A} \quad (4.2.9)$$

式 (4.2.7) 的最大化等价于在解空间选择信号向量 \mathbf{b} 使

$$\begin{aligned} f(\mathbf{b}) &= 2 \int_{-MT_b}^{MT_b+2T_b} S_t(\mathbf{b})y(t)dt - \int_{-MT_b}^{MT_b+2T_b} S_t^2(\mathbf{b})dt \\ &= 2(\mathbf{A}\mathbf{y})^T \mathbf{b} - \mathbf{b}^T \mathbf{A}\mathbf{R}\mathbf{A}\mathbf{b} = 2(\mathbf{A}\mathbf{y})^T \mathbf{b} - \mathbf{b}^T \mathbf{H}\mathbf{b} \end{aligned} \quad (4.2.10)$$

的值最大化。由于使联合最优决策表达式 (4.2.10) 最大化所用的观测值只能通过匹配滤波器输出, 故接收向量 \mathbf{y} 是发送信号 \mathbf{b} 的充分统计量。这个优化函数是设计智能优化检测器的优化目标。

若要求似然函数 $f(\mathbf{b})$ 的最大值解调出来的发送信号 \mathbf{b} 的误码率可以达到理论最小值, 需要遍历所有 $2^{(2M+1)K}$ 个可能解的组合, 从中找出使函数值最大的组合作为检测结果。该方法具有与用户数量和数据长度的乘积呈指数关系的计算复杂度。

Verdu 等在文献[10]中分析了同步 DS-CDMA 系统最优多用户检测器的渐近有效性为

$$\eta_k = \frac{1}{2A_k} \min_{b \in \{-1, +1\}^K} \min_{\substack{d \in \{-1, +1\}^K \\ d_k \neq b_k}} \left| \sum_{i=1}^K A_i b_i s_i(t) - \sum_{i=1}^K A_i d_i s_i(t) \right|^2 = \frac{1}{A_k} \min_{\substack{\mathbf{a} \in \{-1, 0, 1\}^K \\ a_k = 1}} \mathbf{a}^T \mathbf{H} \mathbf{a} \quad (4.2.11)$$

最优多用户检测器的渐近有效性是任何多用户检测器可以达到的最高有效性, 因为当 $\sigma \rightarrow 0$ 时, 该检测器对每一个用户而言都可以达到最小误码率。但大于两个用户的渐近有效性没有定量的公式表达形式。在存在两个用户的情况下, 渐近有效性的理论值可表示为

$$\eta_1 = \min \left[1, 1 + \frac{A_2^2}{A_1^2} - 2\rho_{12} \frac{A_2}{A_1} \right] \quad (4.2.12)$$

$$\eta_2 = \min \left[1, 1 + \frac{A_1^2}{A_2^2} - 2\rho_{21} \frac{A_1}{A_2} \right] \quad (4.2.13)$$

若第 k 个用户的信号线性独立, 令 \mathbf{R}^+ 是归一化互相关矩阵的 Moore-Penrose

广义逆矩阵，则最优多用户检测器的抗远近能力为

$$\bar{\eta}_k = \inf_{\substack{A_i \\ i \neq k}} \eta_k = \frac{1}{R_{kk}^+} \quad (4.2.14)$$

若第 k 个用户的信号线性非独立，则有 $\bar{\eta}_k = 0$ ， R_{kk}^+ 是广义逆矩阵 \mathbf{R}^+ 的第 (k, k) 个元素。

最优多用户检测器的抗远近效应能力是任何一种多用户检测器所能达到的抗远近能力的最小上界，也是任何一种次优检测器性能优劣的一个测度，因此又称最优多用户检测器的抗远近能力为最佳抗远近能力。

4.2.3 Hopfield 神经网络多用户检测器

Hopfield 神经网络多用户检测方法是把最优多用户检测求目标函数的最大值问题转化为求神经网络能量函数的极小值，并通过神经网络状态方程的异步迭代更新来搜索能量函数的极小值。Hopfield 神经网络具有结构简单、易于实现的优点，适合解决多用户检测这类优化问题^[11,12]。但由于采取梯度搜索策略使其成为局部逼近网络，Hopfield 神经网络不能保证总能收敛于全局最优解，其收敛性能受到初始值的影响较大。

若每个神经元用 $u_k (k=1, 2, \dots, n)$ 表示，而 F_k 为该神经元的输入偏置，如果神经元间相连的权值满足 $W_{kk}=0$ ， $W_{kj}=W_{jk}$ 时，则在一个有 n 个神经元的离散 Hopfield 神经网络中，其能量函数

$$E = -\frac{1}{2} \sum_{k=1}^n \sum_{j=1}^n u_k W_{kj} u_j - \sum_{k=1}^n F_k u_k = -\frac{1}{2} \mathbf{u}^T \mathbf{W} \mathbf{u} - \mathbf{F}^T \mathbf{u} \quad (4.2.15)$$

总是单调下降的，并且该神经网络总会收敛到一个稳定状态。

根据多用户检测接收数据的特点和性质，可以进一步把最优多用户检测公式改写为

$$\hat{\mathbf{b}}_{\text{OMD}} = \arg \left\{ \min_{\mathbf{b}} \left[\frac{1}{2} \mathbf{b}^T \mathbf{A}(\mathbf{R} - \mathbf{I}) \mathbf{A} \mathbf{b} - (\mathbf{A} \mathbf{y})^T \mathbf{b} \right] \right\} \quad (4.2.16)$$

其中, $\mathbf{I} = \text{diag}(\mathbf{R})$, 只要令 $\mathbf{W} = -\mathbf{A}(\mathbf{R} - \mathbf{I})\mathbf{A}$, $\mathbf{F} = \mathbf{A} \mathbf{y}$, $\mathbf{u} = \mathbf{b}$, 从而通过求解神经网络的能量函数最小值来得到多用户检测的最优解, 则基于离散 Hopfield 神经网络 (HNN) 的多用户检测器通过离散 Hopfield 神经网络对下式的有限次迭代便可使其快速收敛到相应的稳定状态。

$$u_k(\delta + 1) = \begin{cases} \text{sign}[F_k + \sum_{j=1}^n W_{kj} u_j(\delta)], k = 1 \\ \text{sign}[F_k + \sum_{j=1}^{k-1} W_{kj} u_j(\delta + 1) + \sum_{j=k}^n W_{kj} u_j(\delta)], k = 2 \end{cases} \quad (4.2.17)$$

从表面上看, 似乎 Hopfield 神经网络可以很好地解决多用户检测问题, 但实际应用 Hopfield 神经网络进行优化问题求解时也存在很多问题, 主要有: 神经网络经常会陷入局部极小值而无法收敛到全局最小值, 并且对初值、参数、网络分布和激励函数的要求高, 检测结果鲁棒性较差。为了克服 Hopfield 神经网络的这些缺点, 有学者提出了混沌神经网络和随机神经网络等改进方法。这些方法改善了 Hopfield 神经网络的性能, 但在解决多用户检测问题时依旧不能达到最优检测性能。

4.3 基于免疫克隆量子算法的多用户检测^[13]

在多用户检测设计中, 当前比较有效的人工免疫系统理论是免疫算法理论和免疫进化网络。免疫算法是基于免疫学理论和免疫机制开发的群体模型算法。应用最广泛且与其他演进算法有竞争力的算法是克隆选择算法和免疫算法, 使其与一些智能计算方法互相借鉴和融合, 可有效解决一些工程问题。

本节先根据智能优化算法的特点, 将其和免疫机理^[14]及神经网络的理论有机结合, 给出一种使用神经网络制备疫苗的方法框架, 在该框架下使一些智能优化算法可以快速达到最优多用户检测的性能。然后将量子机制和克隆选择原

理^[15]相结合设计的免疫克隆量子算法,在异步和同步 CDMA 系统进行智能多用户检测器的设计和试验仿真,证明了该理论框架的有效性和可靠性。使用智能计算算法给神经网络提供一个较好的初值,同时将神经网络快速收敛的局部极值提供给智能优化算法,使智能计算算法实现了用较低的计算复杂度获得较优的误码率检测性能。使用随机 Hopfield 神经网络进化量子个体的观测态制备疫苗,使观测态和量子态之间有互动的关系,可以大幅度提高量子进化算法的速度和性能。

4.3.1 神经网络制备疫苗的方法框架

一种有效利用免疫机理的方法框架就是由神经网络制备疫苗,对每次迭代产生的新个体进行疫苗接种。与原智能算法相比,基于免疫机制的智能计算算法所特有的免疫操作主要有疫苗的制作、接种疫苗和接种选择。

(1) 疫苗的制作。随机选出若干优秀个体放入制备疫苗的原料库;在恰当的时机在疫苗原料库中选择个体激活神经网络,激活的神经网络利用智能计算算法提供的良好初值迅速寻找到更好的解,由这些优良解作为母本去制备和抽取疫苗,即以整个个体或个体的绝大部分被抽取作为疫苗准备接种,将其放入制备疫苗的成品库。

(2) 接种疫苗。按一定比例在当前进化的个体集合中随机抽取一定数量的较差个体,并按先前提取的疫苗对这些个体的某些位或所有位进行修改,使所得的新个体以较大的概率接近全局最优解。

(3) 接种选择。对接种了疫苗的个体进行适应度检测,若接种后该个体的适应度下降,说明接种失败,则取消疫苗接种;否则说明接种成功,保留接种的结果。

总结神经网络制备疫苗的结构框架如下:

Step 01 智能算法群体初始化。计算目标函数,选择部分个体放入疫苗原料库,准备制作疫苗。

Step 02 判断算法的迭代终止条件。若是，则结束且输出最优个体；若否，则进行下一步。

Step 03 根据智能计算方法产生新的个体，与此同时选择疫苗原料库中的个体激活神经网络，制备疫苗。

Step 04 计算目标函数，对新产生的个体进行疫苗接种和接种选择，选择优秀个体更新疫苗原料库。

Step 05 判断算法的迭代终止条件。若是，则结束且输出最优个体；否则，返回 **Step 03**。

4.3.2 基于免疫克隆量子算法的多用户检测器

在基于量子计算的优化算法中，一个量子位的状态^[16]可表示为

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (4.3.1)$$

其中， α 和 β 分别是 $|0\rangle$ 和 $|1\rangle$ 的概率幅，且满足下列归一化条件： $|\alpha|^2 + |\beta|^2 = 1$ 。 $|\alpha|^2$ 和 $|\beta|^2$ 分别表示量子态的观测值为 0 和 1 的概率。满足式 (4.3.1) 和归一化条件的一对实数 α 和 β 称为一个量子位的概率幅，记为 $[\alpha, \beta]^T$ 。于是，任意一个标号为 j 且有 n 个量子位的量子个体可表示为

$$\mathbf{Q}_j = [\mathbf{Q}_{j1}, \mathbf{Q}_{j2}, \dots, \mathbf{Q}_{jn}] = \begin{bmatrix} \alpha_{j1}, \alpha_{j2}, \dots, \alpha_{jn} \\ \beta_{j1}, \beta_{j2}, \dots, \beta_{jn} \end{bmatrix} \quad (4.3.2)$$

其中， $|\alpha_{ji}|^2 + |\beta_{ji}|^2 = 1$ ， $-1 \leq \alpha_{ji} \leq 1$ ， $-1 \leq \beta_{ji} \leq 1$ ， $i = 1, 2, \dots, n$ 。若量子逻辑门选用量子旋转门 \mathbf{G} ，它可以表示为

$$\mathbf{G}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (4.3.3)$$

其中， θ 为量子旋转角。若有一量子旋转角为 θ_{ji} ，则第 i 个量子概率幅的更新过程可用下式描述

$$\mathbf{Q}_{ji}^{t+1} = \mathbf{G}(\theta_{ji}^t) \times \mathbf{Q}_{ji}^t \quad (4.3.4)$$

其中, 上标 t 为进化代数, $\mathbf{G}(\theta_{ji}^t)$ 为第 t 代第 i 个量子位的相应量子旋转门, \mathbf{Q}_{ji}^t 为第 t 代的某个体 \mathbf{Q}_j^t 的第 i 个概率幅, \mathbf{Q}_{ji}^{t+1} 为第 $t+1$ 代相应个体 \mathbf{Q}_j^{t+1} 的第 i 个概率幅, \mathbf{Q}_j^t 和 \mathbf{Q}_j^{t+1} 的表现形式可参考式 (4.3.2)。其量子个体的测量态 (或称之为观测态) 通过对量子概率幅测量得到。对于量子概率幅 α_{ji}^{t+1} , 在 $[0,1]$ 间产生均匀随机数, 其值小于 $|\alpha_{ji}^{t+1}|^2$, 则其测量态取 0 (对于多用户检测问题取 -1); 否则, 测量态取 1。

现有的基于量子计算的智能计算方法都通过对量子概率幅的简单测量去获得最优解, 这种方法的最大缺点就是搜索的随机性会产生极大的计算代价。因此, 利用随机 Hopfield 神经网络的快速局部寻优能力, 通过有限次数的更新对测量态进行演进, 再由演进测量态找到需调整量子个体概率幅的位置。若测量态在神经网络作用前后发生了变化, 则说明该位置的概率幅应该向相反方向旋转, 通过量子非门来实现。

$$\begin{bmatrix} \alpha'_{ji} \\ \beta'_{ji} \end{bmatrix} = \mathbf{G}_{\text{NOT}} \begin{bmatrix} \alpha_{ji} \\ \beta_{ji} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha_{ji} \\ \beta_{ji} \end{bmatrix} \quad (4.3.5)$$

随机 Hopfield 神经网络能量函数和最优多用户检测方程的对应关系依旧同 4.2.3 节所介绍的, 可把最优多用户检测问题使用随机 Hopfield 神经网络演进规则进行求解。把随机神经网络处理的观测态及其对应的量子个体制成疫苗放入疫苗库, 在适当的时候进行免疫接种和免疫选择。制作疫苗的关键是激活演进观测态的神经网络。仿真试验表明, 加入各种单调递减的随机扰动的神经元可收敛到更好的局部最优解。综合有关随机 Hopfield 网络各方面的研究成果, 随机 Hopfield 网络每个神经元状态的异步更新规则为

$$u_i(k+1) = \begin{cases} h\{\sum_{j=1}^n W_{ij}u_j(k) + F_i + a(k)z_i(k)\}, l=1 \\ h\{\sum_{j=1}^{l-1} W_{ij}u_j(k+1) + \sum_{j=l}^n W_{ij}u_j(k) + F_i + a(k)z_i(k)\}, l=2 \end{cases} \quad (4.3.6)$$

其中, $a(k)$ 随神经元更新时间的推移而递减, 激活函数 $h(\cdot)$ 和随机扰动函数

$z_i(k)$ 是神经网络中重要的参数, 不同的参数选择对应于不同的网络结构。可使用的激活函数有: (1) 硬限幅函数: $h(x) = \text{sign}(x)$; (2) \tanh 函数: $h(x) = \tanh[c(k)x]$; (3) atan 函数: $h(x) = 2\text{atan}[c(k)x] / \pi$; (4) 高斯函数: $h(x) = e^{-(a_1 x)^2}$, 其中 a_1 为控制参数; (5) sigmoid 函数: $h(x) = 1 / (1 + e^{-a_2 x})$ 。常用的随机扰动分布函数有: (1) 柯西分布; (2) 高斯分布; (3) 均匀分布; (4) Logsig 分布; (5) 混沌机制。所有扰动值的均值应该设计成近似等于 0。

因为要解决多用户检测问题的解是由二进制数字串 $\{-1, +1\}$ 构成的, 只需把量子个体中对概率幅的测量的 0 态映射为信号位 -1, 就可以使用量子优化算法解决多用户检测问题。在新遗传量子算法 (NGQA) 中采用公式完成量子旋转角的更新和方向选择, 并引入量子非门以一定的小概率对量子个体进行翻转操作完成量子个体的变异。算法中第 j 个量子个体的第 i 个量子位的量子旋转角为 $\theta_{ji} = \Delta\theta_{ji}s(\alpha_{ji}, \beta_{ji})$ ($i = 1, 2, \dots, n$), 其方向选择策略如下式所示:

$$s(\alpha_{ji}, \beta_{ji}) = \text{sign}[(f(\mathbf{d}_j) - f(\mathbf{b}'))(d_{ji} - b'_i)\alpha_{ji}\beta_{ji}] \quad (4.3.7)$$

其中, $s(\alpha_{ji}, \beta_{ji})$ 的值可由符号函数 sign 取值为 $\{-1, 0, +1\}$, 该调整策略的思想就是将第 j 个量子个体的当前测量态 $\mathbf{d}_j = [d_{j1}, d_{j2}, \dots, d_{jn}]^T$ 的亲合度 $f(\mathbf{d}_j)$ 与该个体的当前的目标亲合度 $f(\mathbf{b}')$ 进行比较, 调整其第 i 个量子概率幅使测量态向着最有利于找到最优解的方向 (d_{ji} 或 b'_i) 演化, $\mathbf{b}' = [b'_1, b'_2, \dots, b'_n]^T$ 是整个量子抗体集合至今所能得到的最优测量态。

使用随机 Hopfield 神经网络制备疫苗设计多用户检测器的神经网络能量函数为

$$E(\mathbf{b}) = -\frac{1}{2}\mathbf{b}^T \mathbf{A}(\mathbf{I} - \mathbf{R})\mathbf{A}\mathbf{b} - (\mathbf{A}\mathbf{y})^T \mathbf{b} \quad (4.3.8)$$

其中, \mathbf{b} 为待求向量。若采用免疫克隆量子算法来解决此优化问题, 则免疫克隆量子算法的目标函数 (亲合度函数) 进一步可写为

$$f(\mathbf{b}) = K_w + [-E(\mathbf{b}) + E_w] \quad (4.3.9)$$

式中, $-E_w$ 是首代中最差量子抗体 (个体) 所得到的能量函数的负值, K_w 为

一个正常数。所提算法首先结合新遗传量子算法和克隆选择算法的优化机制和结构,设计兼有二者优点的克隆量子算法,然后在克隆量子算法中引入免疫操作,随机神经网络参与制备疫苗。提出基于免疫克隆量子算法(ICQA)多用户检测器的主要步骤可以描述为:

Step 01 抗体集合初始化。确定量子抗体集合大小 N_p 和量子抗体的量子位数 n , 包含 N_p 个量子抗体的集合 $\mathbf{Q} = \{\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_{N_p}\}$ 。其中, $\mathbf{Q}_j (j=1, 2, \dots, N_p)$ 为抗体集合的第 j 个量子抗体; 所有的初始量子概率幅均取 $1/\sqrt{2}$, 表示在初始时抗体的所有状态以相同的概率进行叠加。

Step 02 根据 \mathbf{Q} 中各个量子抗体的概率幅构造出量子叠加态的观测态 \mathbf{D} (为加快收敛速度, 可取传统检测器的输出作为一个初始观测状态), $\mathbf{D} = \{d_1, d_2, \dots, d_{N_p}\}$, 其中 d_j 为第 j 个量子抗体的观测状态, 即一个二进制串。随机选取 N_r 个量子抗体及其观测态放入疫苗库作为制作疫苗的备用抗体, 准备制作疫苗。

Step 03 计算亲和度函数, 对抗体集合中的所有抗体进行亲和度评价, 判断算法是否满足迭代终止条件 (设为最大迭代次数)。若是, 则算法终止且输出最优抗体; 否则, 执行下一步。

Step 04 选择操作压缩生成 \mathbf{Q}_b (压缩后抗体为 N_q 个), 保留最佳观测抗体作为演进目标。

Step 05 疫苗的制备。从疫苗库中提取备用的量子抗体及观测态, 随机选择部分抗体, 激活随机 Hopfield 网络对抗体 (个体) 的观测态进行演进更新, 比较观测态更新前后的位值变化, 有变化说明量子抗体的概率幅需反转, 则其相应的量子概率幅通过量子非门完成倒置操作。

Step 06 克隆 \mathbf{Q}_b 生成 \mathbf{Q}_c , 恢复量子抗体集合的群体规模。

Step 07 根据 \mathbf{Q}_c 更新生成 \mathbf{Q}_d 。先确定量子旋转角, 再确定量子旋转门并作用于抗体集合中的所有抗体。然后对克隆的量子抗体概率幅以一定的概率 ($\frac{1}{n} \sim \frac{3}{n}$) 使用量子非门对量子概率幅进行反转, 生成 \mathbf{Q}_d 。

Step 08 产生观测态，进行亲和度计算，选择 N_r 个量子抗体和观测态放入疫苗库，更新制备疫苗的母本，进行免疫接种和接种选择。

Step 09 判断算法是否满足迭代终止条件。若是，则算法终止；否则，迭代次数增 1，算法转至 **Step 04** 继续执行。

4.3.3 试验仿真

假设有 K 个用户的 DS-CDMA 通信系统，采用 31 位的 Gold 序列作为扩频序列，扩频序列最大的归一化互相关系数为 $9/31$ 。在仿真过程中，使用的多用户检测器有：传统检测器（CD）；基于遗传算法（GA）的多用户检测器^[17]；基于遗传量子算法（GQA）^[16]的多用户检测器；基于 Hopfield 神经网络（HNN）的多用户检测器^[18]；基于随机 Hopfield 神经网络（SHNN）的多用户检测器^[19]；基于新的遗传量子算法（NGQA）的多用户检测器^[1]；基于免疫克隆量子算法（ICQA）的多用户检测器；最优多用户检测器（OMD）。为了便于比较智能算法的收敛速度和收敛性能，GA、GQA、NGQA 和 ICQA 的种群空间所含个体数都设为 N_p 。在 ICQA 中旋转步长取 $\Delta\theta_{ji} = 0.03\pi$ 。

在同步 CDMA 系统中进行仿真，设定用户 1 为目标用户，且 $K = N_p = 10$ ， $N_q = 5$ ， $N_r = 3$ 。最大迭代次数设为 50，10 个用户的信号能量 E_b 相等，其信噪比 E_b / N_0 为 5dB 时，GA、GQA、NGQA 和 ICQA 的误码率和迭代次数的关系如图 4.3.1 所示。在信号能量 E_1 保持单位能量不变且信噪比 $E_1 / N_0 = 3$ dB，并且其他用户 2~10 的信号能量 E_i 和目标用户的能量比（远近比）为 $E_i / E_1 = -4$ dB 时，用户 1 的误码率随迭代次数变化的曲线如图 4.3.2 所示。从图 4.3.1 和图 4.3.2 中可以看出，ICQA 比 GA 和 GQA 有较快的初始收敛速度和较好的全局收敛性能；ICQA 在第 3 代的误码率性能就超过 GA 和 GQA 在第 50 代的误码率性能，并且收敛性能已经接近最优多用户检测器。

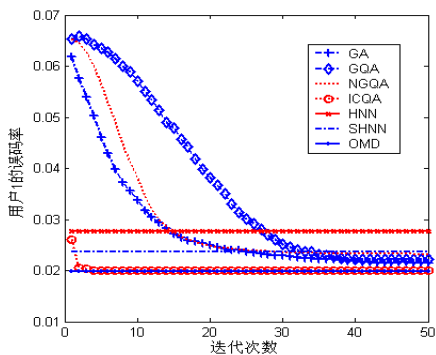


图 4.3.1 多址干扰存在时收敛性比较曲线

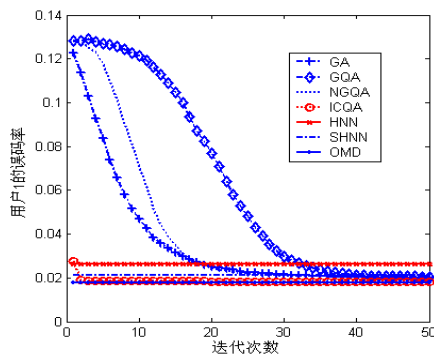


图 4.3.2 远近效应存在时收敛性比较曲线

在考察各算法抗多址干扰的能力和抗远近效应的能力时, GA、GQA 和 NGQA 的最大迭代次数设为 10, ICQA 的最大迭代次数仅设为 5。在严格功率控制下, 10 个用户的信号能量相等, 选择每一用户的信噪比从 2dB 增加到 10dB 来检验所选用的 8 种检测器在不同信噪比下的误码率, 所得结果如图 4.3.3 所示。设用户 1 为目标用户, 其信号能量 E_1 保持单位能量不变且其信噪比固定在 5dB, 其他用户 2~10 的信号能量 E_i 进行变化以得到不同的远近比 E_i / E_1 , 通过用户 1 的误码率来检验所给出的 8 种检测器的抗远近效应能力, 得到的仿真结果如图 4.3.4 所示。

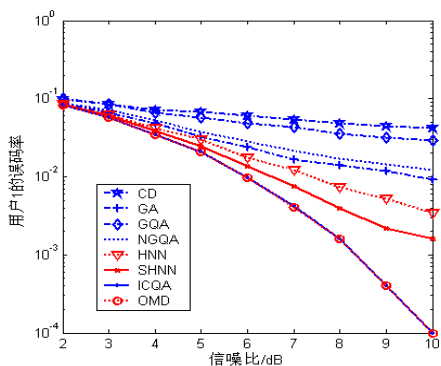


图 4.3.3 用户 1 误码率和信噪比关系曲线

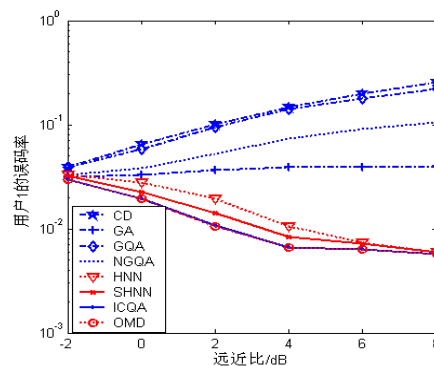


图 4.3.4 用户 1 误码率和远近比关系曲线

图 4.3.3 和图 4.3.4 证明了 ICQA 在迭代 5 次时的抗多址干扰和抗远近效应能力远优于 CD、GQA、HNN、SHNN、NGQA 和 GA, 达到了 OMD 的收敛性

能。从图 4.3.3 和图 4.3.4 中可以看出, GQA、NGQA 和 GA 在迭代 10 次时的抗多址干扰和抗远近效应能力依旧很差, 无论其收敛速度还是收敛性能都不如所提出的 ICQA 算法。

在对异步 CDMA 系统进行仿真时, 设 $N_q = 5$, $N_r = 3$, $N_p = 10$, 设定用户数为 $K = 4$, 每个用户传递的一帧码元数量 $P = 3$ 。令 $\tau_i - \tau_{i+1}$, $\tau_{i+1} - \tau_i = 2T_c$ 。图 4.3.5 作出了信噪比为 5dB 时用户 1 的误码率随迭代次数变化的曲线, 图 4.3.6 作出了 $E_i / N_0 = 2 \text{ dB}$ 和 $E_i / E_1 = 4 \text{ dB}$ 时用户 1 的误码率随迭代次数变化的曲线, ICQA 可在最短的时间达到 OMD。

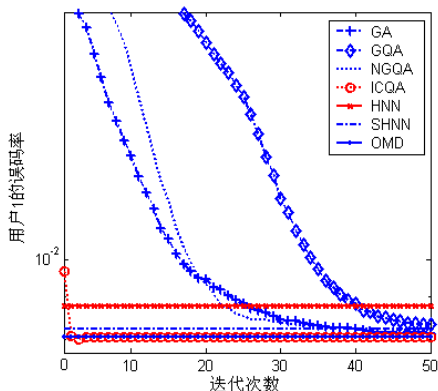


图 4.3.5 多址干扰存在时收敛性比较曲线

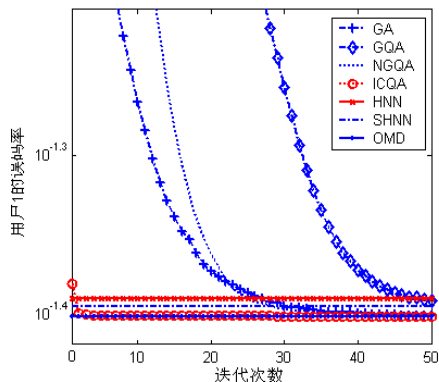


图 4.3.6 远近效应存在时收敛性比较曲线

4.4 量子 Hopfield 神经网络的多用户检测设计

基于神经网络的多用户检测方法是一种可快速实现, 且在一定程度上能克服多址干扰和远近效应影响的多用户检测方法^[20,21]。由于人工神经网络具有自学习、联想记忆、自适应和非线性逼近等功能, 人工神经网络设计的多用户检测器通过训练后可以在不具备对象先验知识的条件下, 仅根据输出数据就可直接进行检测, 它为解决复杂的多用户检测问题提供了一条途径。然而, 随着应用的深入和实际问题的日益涌现, 基于训练序列的神经网络多用户检测器

的局限和不足也逐渐显现出来^[22]，存在收敛性能差和收敛结果受训练过程的限制等缺点。为了克服神经网络的局限和不足，国内外很多专家进行了深入研究，提出了很多有意义的改进措施，如改进训练算法、改进网络结构及改进神经网络结构等^[23-26]。随着量子计算逐渐显现出强大的能力，很多学者开始考虑将量子理论和神经网络相结合，在本质上改善神经网络的计算性能，这就是量子神经网络^[27,28]，但这些需要训练的量子神经网络，不适应复杂噪声环境下的信号检测。本节就在深入研究离散 Hopfield 神经网络的基础上，对量子计算进行深入研究，介绍了一种求解多用户检测问题的量子神经网络计算模型。该模型利用量子旋转门对神经网络的神经元进行更新，并且充分利用量子神经网络可测量功能，能进一步提高量子神经网络的检测性能，并且该量子神经网络无须训练，适合多用户检测问题的快速求解。

4.4.1 量子 Hopfield 神经网络

离散量子神经网络的结构可以用图 4.4.1 表示，它有 n 个量子神经元通过相互连接形成，每个量子神经元都有一个量子外部输入 G_i ，每个神经元输出信号有 $\{-1, +1\}$ 两种可能的状态。量子神经元通过量子态进行演进更新，其中的任意量子神经元 i 与 j 之间的突触权值为 $w_{ij} = w_{ji}$ ， $w_{ii} = 0$ 。这样的权值表明每个量子神经元与自己并无连接，但每个量子神经元都同其他量子神经元通过权值相连，即每个量子神经元都将其输出通过权值传递给其他的神经元，同时每个量子神经元又接收其他量子神经元传来的信息，从而确定出相应的量子旋转角，通过量子旋转门实现量子神经元状态的更新。这样对于每个量子神经元来说，其输出信号经过其他量子神经元联合作用后反馈给自己，所以说量子 Hopfield 网络是一种反馈型的量子神经网络。

在图 4.4.1 中，各量子神经元有相同的结构，各节点有相同的转移特性函数。网络输入为量子态 $\bar{u} = [u_1^T, u_2^T, \dots, u_n^T]^T = \begin{bmatrix} u_1, u_2, \dots, u_n \\ \beta_1, \beta_2, \dots, \beta_n \end{bmatrix}^T$ ， $-1 \leq u_i \leq 1$ ， $-1 \leq \beta_i \leq 1$ ， $i = 1, 2, \dots, n$ 。量子神经网络输出为二进制状态 $v = [v_1, v_2, \dots, v_n]^T$ ，量子

神经元的外部输出又称作量子偏置，可以表示为 $\mathbf{G}_i = [I_i, \sqrt{1-I_i^2}]^T$ ，则第 i 个量子神经元在 t 时刻的状态为 $v_i(t)$ ($t=0,1,2,\dots$)。当量子神经元 i 输入为 $\mathbf{u}_i = [u_i, \beta_i]$ ，量子旋转角度为 $\varphi_i^{t+1} = -\sum_{j=1, j \neq i}^n w_{ij} v_j$ 时，若 φ_i^{t+1} 超出 $[-\pi/2, \pi/2]$ 的范围，

则需要将其限制在边界上。

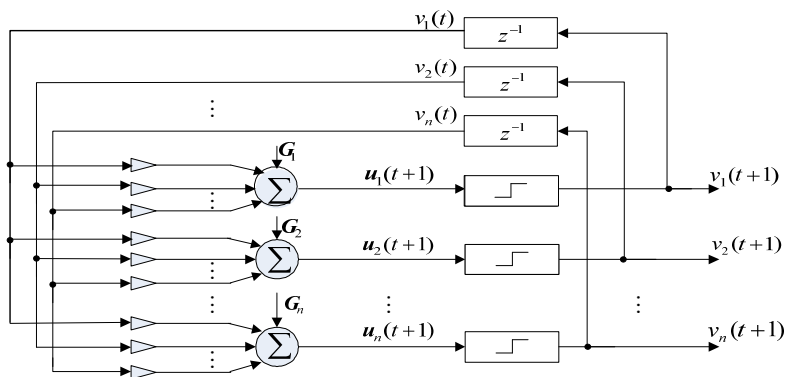


图 4.4.1 量子 Hopfield 神经网络结构

$$[\mathbf{u}_i(t+1)]^T = \mathbf{U}(\varphi_i^{t+1}) \mathbf{G}_i = \begin{bmatrix} \cos \varphi_i^{t+1} & -\sin \varphi_i^{t+1} \\ \sin \varphi_i^{t+1} & \cos \varphi_i^{t+1} \end{bmatrix} \mathbf{G}_i \quad (4.4.1)$$

对量子神经元的实部进行判决就得到神经元的输出为

$$v_i(t+1) = \text{sgn}[\text{Re}((\mathbf{u}_i(t+1))^T)] = \text{sgn}[u_i(t+1)] = \begin{cases} +1, & u_i(t+1) > 0 \\ -1, & u_i(t+1) < 0 \end{cases} \quad (4.4.2)$$

量子神经网络一般有串行异步更新和并行同步更新两种运行方式。由于采用同步更新运行的量子神经网络，其达到一个收敛的状态需要很严格的条件，这在多用户检测中难以得到满足，所以要采用异步更新的方式。即在任意时刻，只对某一量子神经元 i （随机或以确定的规则选择）的状态更新，而其他量子神经元的状态保持不变。量子神经网络的运行过程描述如下：

Step 01 确定量子偏置状态（外部量子输入），量子神经网络初始化。

Step 02 从量子神经网络中随机（或依次）选择量子神经元。

Step 03 确定量子旋转角，按式（4.4.1）求出该神经元的输入 $\mathbf{u}_i(t+1)$ 。

Step 04 按式 (4.4.2) 求出该量子神经元的输出 $v_i(t+1)$ ，此时网络中其他量子神经元的输出保持不变。

Step 05 判断量子神经网络是否达到收敛的条件。若是，就停止；否则，转至 **Step 02** 继续执行。

若量子神经网络从某一次迭代以后，其输出的所有状态不再发生变化，就定义量子神经网络处于稳定状态。任意一个神经元的稳定输出状态用公式表示为 $v_i(t) = v_i(t+1) = \dots = v_i(\infty)$ 。对于某一优化问题，若离散 Hopfield 神经网络的能量函数为 $E = -\frac{1}{2} \mathbf{v}^T(t) \mathbf{W} \mathbf{v}(t) - \mathbf{F}^T \mathbf{v}(t)$ ，设 $\mathbf{w} = \mathbf{W} / \max\{|F_1|, |F_2|, \dots, |F_n|\}$ ， $\mathbf{I} = \mathbf{F} / \max\{|F_1|, |F_2|, \dots, |F_n|\}$ ，则量子神经网络能量函数就可以定义为

$$\bar{E} = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n v_i(t) w_{ij} v_j(t) - \sum_{i=1}^n I_i v_i(t) = -\frac{1}{2} \mathbf{v}^T(t) \mathbf{w} \mathbf{v}(t) - \mathbf{I}^T \mathbf{v}(t) \quad (4.4.3)$$

其中， $w_{ij} = w_{ji}$ ， $w_{ii} = 0$ 。根据经典 Hopfield 神经网络的理论和量子计算的特点可以得出结论：量子神经网络容易收敛到一个稳定的状态。

4.4.2 基于量子神经网络的多用户检测

由于最优多用户检测方程经过映射可以得到量子神经网络能量函数的形式，因此可以使用其设计最优多用户检测器，结合量子神经网络的可测量特性，下面介绍异步更新方式的量子神经网络的运行步骤。

Step 01 将要解决的最优多用户检测问题映射到量子空间。对两个量子神经网络初始化，把传统检测器的判决输出作为两个量子神经网络的初始二进制输入。

Step 02 两个量子神经网络依次选取量子神经元和量子偏置状态。

Step 03 确定每个量子神经网络的量子旋转角。一个量子神经网络的旋转角为 $\varphi_i^{t+1} = -\sum_{j=1, j \neq i}^n w_{ij} v_j(t)$ ，另一个量子神经网络的旋转角为 $\varphi_i^{t+1} = -\sum_{j=1, j \neq i}^n w_{ij} v_j(t) +$

$\text{randn} / (it)$ ，其中 randn 为均值为 0、方差为 1 的高斯随机数。若 φ_i^{t+1} 超出 $[-\pi/2, \pi/2]$ 的范围，则需要将其限制在边界上，然后分别按式 (4.4.1) 求出每个量子神经元的输入 $u_i(t+1)$ 。

Step 04 按式 (4.4.2) 求出量子神经元的输出 $v_i(t+1)$ ，此时网络中其他量子神经元的输出保持不变。

Step 05 判断量子网络是否达到稳定状态，若否则转至 **Step 02**，若是则执行下一步。

Step 06 计算量子神经网络输出能量，对两个量子神经网络的量子态进行多次测量或对测量态执行小概率变异操作，找到测量态的当前最优解。

Step 07 判断是否达到迭代终止条件（一般可设执行 **Step 06** 的最大次数为 2~3 次即可），若是则输出当前最优解结束，否则把测量态的当前两个最好的解分别作为两个量子神经网络的初态，返回 **Step 02** 继续演进。

4.4.3 基于量子神经网络的多用户检测器仿真

在考察各算法的抗多址干扰能力的时候，设定所有用户的信号功率相等且为 E_b ，则传统检测器 CD、HNN 和 QNN 检测器的目标用户误码率和信噪比的关系如图 4.4.2 所示。在考察算法的抗远近效应的能力时，用户 1 的信号能量 E_1 保持单位能量不变，同时假设信噪比 E_1 / N_0 固定在 4.5dB，其他用户为干扰用户的信号能量相同且为 E_i ，则 CD、HNN、QNN 和 OMD 的误码率和远近比的关系如图 4.4.3 所示。图 4.4.2 和图 4.4.3 证明了量子神经网络设计的多用户检测器的抗多址干扰和抗远近效应能力远优于 CD 和 HNN，但是依旧没有达到 OMD 的性能。若要进一步提高性能，需要对量子神经网络进行多次测量后选择优秀解，再使用量子神经网络进行多次演进。

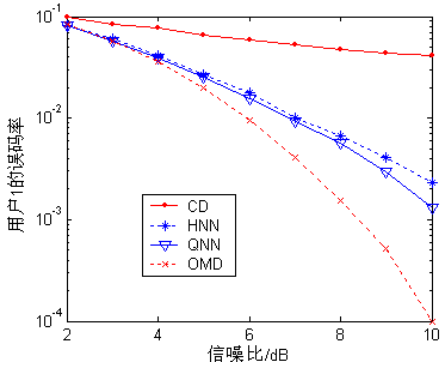


图 4.4.2 误码率和信噪比关系曲线

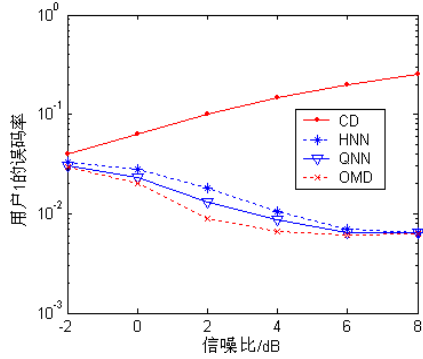


图 4.4.3 误码率和远近比关系曲线

4.5 量子蜂群算法的鲁棒多用户检测

过去的 20 年,众多学者在高斯白噪声条件下进行多用户检测技术研究,取得了丰硕成果,可以有效地抵抗多址干扰和远近效应^[29]。然而,在实际应用中,由于复杂的电磁干扰和广泛存在的具有脉冲特性的自然噪声,其噪声不再满足高斯分布,具有冲击噪声特性。就系统容量而言,使用传统的多用户检测方法,脉冲信道的冲击性越大,其所能容纳的用户数也就越少。若不设计专门的检测器,冲击噪声能使通信系统性能恶化严重。因此,研究冲击噪声特性,设计鲁棒多用户检测器(Robust Multiuser Detector, RMD)使其在冲击噪声背景下有最优性能将有更广泛的应用价值。

众所周知,高斯噪声背景下的 CDMA 系统的最优多用户检测是典型的组合优化问题,一些离散优化算法如神经网络粒子群算法^[30]、克隆遗传量子算法^[31]、遗传退火禁忌搜索算法^[32]和粒子群优化^[33]等智能算法被用来解决 CDMA 多用户检测问题。现有研究已经表明,在高斯噪声环境下,优化函数结构的特殊性使得神经网络和智能算法相结合可在很小计算量的情况下达到最优检测性能^[34,35]。

但由于冲击噪声环境下的多用户检测优化函数没有特殊性,故不能把高斯噪声环境下的多用户检测算法直接进行移植。在现有计算条件下,冲击噪声环

境下的多用户检测技术根据穷尽搜索办法是不能在有限的时间内实时实现的，因此根据鲁棒多用户检测问题的现实需要去设计简单高效的新算法是值得努力的目标。

蜂群优化算法是近年来被大家密切关注的智能优化算法，由于其应用的广泛性和良好的性能，被量子化改进并应用到通信领域。为了简化量子蜂群算法^[36]的计算量，有效解决冲击噪声环境下的多用户检测问题，本节使用单链离散量子蜂群算法去设计鲁棒多用户检测器。所提的量子蜂群算法根据蜂群机理演化量子旋转角，使用模拟量子旋转门更新每个蜜蜂的量子位置，可高效求得冲击噪声环境下的多用户检测问题的最优解。

4.5.1 鲁棒多用户检测器

设有 K 个用户的 CDMA 系统，在某时刻的接收信号为

$$\mathbf{r} = \sum_{k=1}^K A_k b_k \mathbf{s}_k + \mathbf{n} = \mathbf{SAb} + \mathbf{n} \quad (4.5.1)$$

式中， $\mathbf{s}_k = \frac{1}{\sqrt{N}}[s_{1k}, s_{2k}, \dots, s_{Nk}]^T$ ， $s_{jk} \in \{-1, +1\}$ 是用户 k 的归一化特征波形； N 是处理增益； $b_k \in \{-1, +1\}$ 和 A_k 分别是用户 k 的数据比特和信号幅值； $\mathbf{A} = \text{diag}(A_1, \dots, A_K)$ ； $\mathbf{b} = [b_1, b_2, \dots, b_K]^T$ ； $\mathbf{n} = [n_1, n_2, \dots, n_N]^T$ 代表冲击噪声向量，它们各分量独立同分布。对于冲击噪声，可用特征指数 α 代表分布拖尾的严重程度，且 $0 < \alpha < 2$ ，值越小拖尾越严重，而噪声离差 γ 又称为散射参数。

鲁棒多用户检测问题可由误差平方的最小化求得，即

$$\hat{\mathbf{b}}_{LS} = \arg \min_{\mathbf{b}} \sum_{j=1}^N (r_j - \sum_{k=1}^K s_{j,k} A_k b_k)^2 = \arg \min_{\mathbf{b}} \|\mathbf{r} - \mathbf{SAb}\|^2 \quad (4.5.2)$$

最优鲁棒 Huber 多用户检测器是其性能对实际模型微小变化不敏感的检测器。应用 Huber 理论，式 (4.5.2) 可利用残差的非快增函数 ρ 之和的最小化表示为

$$\hat{b} = \arg \min_b \sum_{j=1}^N \rho(r_j - \sum_{k=1}^K s_{jk} A_k b_k) \quad (4.5.3)$$

Minmax M 估计器的 Huber 罚函数为

$$\rho(y_j) = \begin{cases} \frac{y_j^2}{2z_j^2}, & |y_j| \leq \varepsilon_j z_j^2 \\ \varepsilon_j |y_j|^2 - \frac{\varepsilon_j^2 z_j^2}{2}, & |y_j| > \varepsilon_j z_j^2 \end{cases} \quad (4.5.4)$$

可以取 $\varepsilon_j = \frac{3}{2z_j}$ ，冲击噪声环境下 z_j 值的大小与离差 γ_j 有关，可以设 $\gamma_j = \sqrt{2}z_j$ 。

4.5.2 新量子蜂群算法

若在量子蜂群中有 h 只量子蜜蜂，第 i 只量子蜜蜂的量子位置可以表示为

$$\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD}) \quad (4.5.5)$$

其中， $0 \leq v_{id} \leq 1$ ， $d = 1, 2, \dots, D$ 。

设 θ_{id}^{t+1} 为第 $t+1$ 次迭代的一个量子旋转角，量子比特 v_{id}^t 用模拟量子旋转门进行演进，第 i 只量子蜜蜂量子位置的第 d 个量子位的更新如下

$$v_{id}^{t+1} = |v_{id}^t \cos(\theta_{id}^{t+1}) - \sqrt{1 - (v_{id}^t)^2} \sin(\theta_{id}^{t+1})| \quad (4.5.6)$$

量子蜂群包含两种蜜蜂：量子工蜂和量子观察蜂。每一只量子蜜蜂在每一次循环中都会找到一个食物源，并且其根据自己的历史经验和整个蜂群的经验，在位置空间中根据量子计算和群体智能演进规则找寻蜜源含量最优秀的位置。

在量子蜂群中，标号 $i = 1, 2, \dots, \frac{h}{2}$ 的蜜蜂为量子工蜂，标号 $i = \frac{h}{2} + 1, \frac{h}{2} + 2, \dots, h$ 的蜜蜂为量子观察蜂。设 $\mathbf{v}_i^t = [v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t]$ ($i = 1, 2, \dots, h$) 为第 i 只量子蜜蜂的量子位置，对量子位置进行测量得到量子蜜蜂的位置，也是蜜源的位置，可表示为 $\mathbf{x}_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t]^T$ ($i = 1, 2, \dots, h$)，其可用 D 维取值 $\{-1, 1\}$ 的数字串代

表。到目前为止第 i 只量子蜜蜂搜索到的局部最优位置可以表示为 $\mathbf{p}_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t]^\top, i=1, 2, \dots, h$ 。到现在为止整个量子蜂群所搜索到的全局最优位置可以表示为 $\mathbf{p}_g^t = [p_{g1}^t, p_{g2}^t, \dots, p_{gD}^t]^\top$ 。

在第 $t+1$ 次迭代中, 第 i 只量子工蜂的第 d 维量子旋转角为

$$\theta_{id}^{t+1} = e_1(p_{id}^t - x_{id}^t) + e_2(p_{gd}^t - x_{id}^t) \quad (4.5.7)$$

其中, $i=1, 2, \dots, h/2, d=1, 2, \dots, D, p_{id}^t$ 和 p_{gd}^t 对量子旋转角的影响程度分别由影响因子 e_1 和 e_2 决定, $t+1$ 和 t 代表了迭代次数。

第 $i (i=h/2+1, h/2+2, \dots, h)$ 只量子观察蜂根据轮盘赌的方法选择量子工蜂参与量子旋转角的演进。设第 j 只量子工蜂局部最优位置的适应度为 $fit(\mathbf{p}_j)$, 构造适应度函数并使其值必大于 0, 则第 $j (j=1, 2, \dots, h/2)$ 只量子工蜂被选择的概率为

$$q_j^{t+1} = \frac{fit(\mathbf{p}_j^t)}{\sum_{k=1}^{h/2} fit(\mathbf{p}_k^t)} \quad (4.5.8)$$

在第 $t+1$ 次迭代中, 第 i 只量子观察蜂的第 d 维量子旋转角为

$$\theta_{id}^{t+1} = e_3(p_{jd}^t - x_{id}^t) + e_4(p_{id}^t - x_{id}^t) + e_5(p_{gd}^t - x_{id}^t) \quad (4.5.9)$$

其中, $i=h/2+1, h/2+2, \dots, h, d=1, 2, \dots, D, p_{jd}^t, p_{id}^t$ 和 p_{gd}^t 对量子旋转角的影响程度分别由影响因子 e_3, e_4 和 e_5 决定。

第 i 只量子蜜蜂的第 d 位量子比特位置根据量子旋转角进行如下更新

$$v_{id}^{t+1} = \begin{cases} \sqrt{1-(v_{id}^t)^2}, & \theta_{id}^{t+1}=0 \text{ 且 } \eta_{id}^{t+1} < c_1 \\ |v_{id}^t \cos(\theta_{id}^{t+1}) - \sqrt{1-(v_{id}^t)^2} \sin(\theta_{id}^{t+1})|, & \text{其他} \end{cases} \quad (4.5.10)$$

其中, $d=1, 2, \dots, D; \eta_{id}^{t+1}$ 代表随机数, 它在 $[0, 1]$ 之间均匀分布; 而 c_1 代表当量子位的量子旋转角为 0 时对该量子位使用量子非门进行翻转操作, 其值为 $[0, 1/D]$ 间的常量, 根据不同的问题试验选择确定。

第 i 只量子蜜蜂的食物源位置可以对量子位置各量子位的测量得到。

$$x_{id}^{t+1} = \begin{cases} 1, \lambda_{id}^{t+1} > (v_{id}^{t+1})^2 \\ -1, \lambda_{id}^{t+1} < (v_{id}^{t+1})^2 \end{cases} \quad (4.5.11)$$

其中, $d=1,2,\dots,D$, λ_{id}^{t+1} 为 $[0,1]$ 之间均匀分布的随机数, $(v_{id}^{t+1})^2$ 描述了量子位 v_{id}^{t+1} 出现“0”状态的概率。

量子蜜蜂局部最优位置 p_i^t 更新通过下式

$$p_i^{t+1} = \begin{cases} x_i^{t+1}, f(p_i^t) < f(x_i^{t+1}) \\ p_i^t, f(p_i^t) < f(x_i^{t+1}) \end{cases} \quad (4.5.12)$$

最大值优化问题的全局最优位置更新公式为

$$p_g^{t+1} = \arg \max_{p_i^{t+1}} f(p_i^{t+1}) \quad 1 \leq i \leq h \quad (4.5.13)$$

4.5.3 基于量子蜂群算法的鲁棒多用户检测^[37]

若 \mathbf{b} 为待求向量, 可看作量子蜜蜂所找到的食物源位置, 则蜜源含量计算函数为

$$f(\mathbf{b}) = 1 / \left(\sum_{j=1}^N \rho(r_j) - \sum_{k=1}^K s_{jk} A_k b_k \right) + E \quad (4.5.14)$$

其中 $E=0.1$ 。为了满足量子观察蜂轮盘赌选择的需要, 构造适应度函数为

$$fit(\mathbf{b}) = f(\mathbf{b}) \quad (4.5.15)$$

则将鲁棒多用户检测求解问题转化为用量子蜂群算法求解使蜜源含量最佳（此问题为蜜源值最大）的食物源位置。总结鲁棒多用户检测的量子蜂群算法迭代过程如下：

Step 01 初始化量子位置、位置和局部最优位置, 每个量子位置的量子位开始阶段都设置为 $1/\sqrt{2}$ 。对量子蜜蜂的量子位进行测量操作可获得量子蜜蜂的初始位置, 同时把量子蜜蜂的初始位置设置为其局部最优位置。

Step 02 使用蜜源含量函数获得每个位置的蜜源量, 挑选出蜜源量最大位置

为整个量子蜂群的全局最优位置。

Step 03 更新每个量子工蜂的量子位置和位置。

Step 04 每个量子观察蜂根据轮盘赌选择原理选择工蜂的局部最优位置，然后更新其量子位置和位置。

Step 05 根据蜜源含量公式计算每个量子蜜蜂的新位置的蜜源含量。

Step 06 对每个量子蜜蜂，若新位置优于其局部最优位置，则其局部最优位置更新，把局部最优位置的最大值作为全局最优位置。

Step 07 如果迭代次数未达到最大迭代次数，则返回 **Step 03**；否则，结束迭代，输出量子蜜蜂的全局最优位置。

4.5.4 试验仿真

假设在 DS-CDMA 通信系统中，环境噪声为冲击噪声，用于扩频的 Gold 序列码的最大的归一化互相关系数为 9/31，以验证恶劣干扰下多用户检测器的性能。仿真使用所设计算法和已有的三种智能算法进行比较，分别是遗传算法、粒子群优化和量子遗传算法，所对应的鲁棒多用户检测器分别记作 GA-RMD、PSO-RMD 和 QGA-RMD；基于所提的量子蜂群算法的鲁棒多用户检测器记作 NQBC-RMD。

GA-RMD、PSO-RMD、QGA-RMD 和 NQBC-RMD 的种群中个体数都设为 16。在仿真过程中，遗传算法(GA)、量子遗传算法(QGA)和粒子群算法(PSO)的其他参数设置参考文献[38]和[39]设定。对于新量子蜂群优化算法(NQBC)，参数设置如下： $e_1 = 0.03\pi$ ， $e_2 = 0.015\pi$ ， $e_3 = 0.015$ ， $e_4 = 0.03$ ， $e_5 = 0.005$ ， $c_1 = 0.1/D$ ， D 为搜索空间维数。

第 i 个用户的广义信噪比定义为 $GSNR_i = 10 \lg \frac{A_i^2}{2\gamma_i^2}$ ， γ_i 为第 i 个用户的噪声

离差。为了在相同种群规模和相同迭代次数条件下比较 4 种优化算法的性能和

收敛速度, 将特征指数设定为 1.6。设所有用户的信号功率相等, 当广义信噪比是 5dB 的时候, 图 4.5.1 给出了 4 种智能算法实现的鲁棒多用户检测的误码率收敛性能曲线。将用户 1 的广义信噪比固定在 4dB, 当用户 1 与其他用户功率的远近比是 3dB 的时候, 图 4.5.2 给出了 4 种智能算法实现的鲁棒多用户检测的误码率收敛性能曲线。

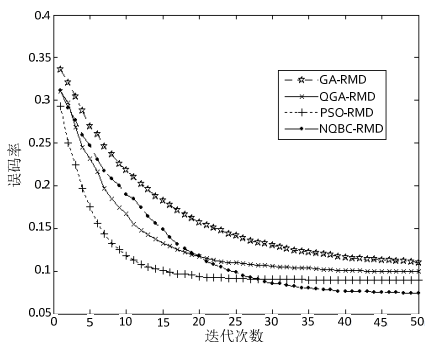


图 4.5.1 4 种算法误码率和迭代次数关系比较

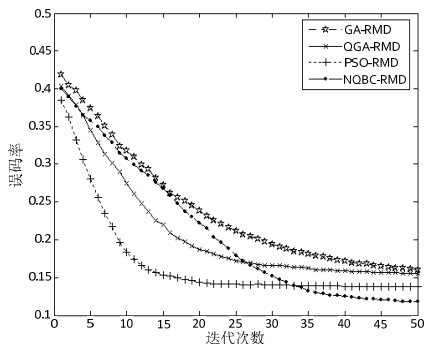


图 4.5.2 4 种算法用户 1 误码率和迭代次数关系比较

从图 4.5.1 和图 4.5.2 中可以看到, NQBC-RMD 的收敛速度和收敛性能都优于 PSO-RMD、GA-RMD 和 QGA-RMD。当特征指数取 1.6 时, 16 个用户功率相等, 误码率和广义信噪比关系曲线如图 4.5.3 所示。设用户 1 的广义信噪比为 5dB, 其他用户功率相等, 则用户 1 的误码率和远近比关系曲线如图 4.5.4 所示。

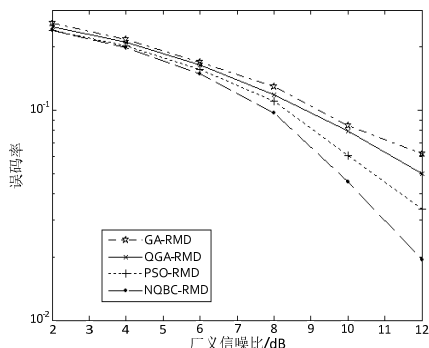


图 4.5.3 4 种算法误码率和广义信噪比关系比较

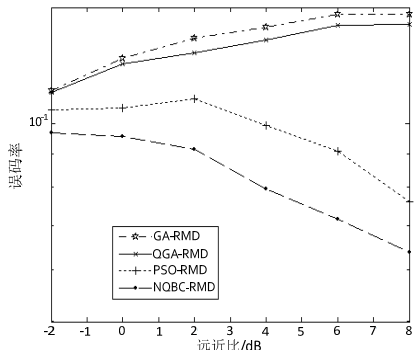


图 4.5.4 4 种算法误码率和远近比关系比较

根据图 4.5.3 和图 4.5.4, 在相同的仿真环境下, 基于 4 种智能计算算法的鲁棒多用户检测器性能差异很大, 所设计的 NQBC-RMD 优于 PSO-RMD、GA-RMD 和 QGA-RMD, 具有最好的抵抗多址干扰、远近效应和冲击噪声能力。

K 个用户的鲁棒多用户检测穷尽搜索的计算量为 $O(2^K)$, 所设计的 NQBC-RMD 算法的种群规模为 K , 其终止迭代次数为 aK ($a < K$), 其计算量从穷尽搜索的指数级下降到平方级 $O(K^2)$ 。因此, 用户数越多, 所设计算法的计算量节省越大。

4.6 小结

本章主要提出了三种可求解多用户检测问题的量子智能算法。首先在高斯白噪声背景下设计了基于所提框架免疫克隆量子算法的多用户检测结构框架, 并对已有的多用户检测器和所设计的多用户检测算法进行了仿真, 实验证明所设计的免疫克隆量子多用户检测器有很快的收敛速度和使用较少的计算量就达到最优多用户检测的性能。其次, 所设计的量子神经网络多用户检测器可快速收敛到一个满意解。最后, 在冲击噪声环境下设计的基于量子蜂群算法的鲁棒多用户检测器有好的收敛速度和收敛性能。

参考文献

- [1] 高洪元. 多用户检测中的智能信息理论研究. 哈尔滨工程大学博士论文, 2010.
- [2] 王焱滨. 若干计算智能方法在 CDMA 多用户检测中的应用研究. 电子科技大学博士论文, 2003: 13-18, 72-75.
- [3] 高洪元, 刁鸣, 王冰. 基于免疫克隆选择算法的多用户检测技术研究. 系统仿真学报, 2007, 19 (5): 983-986.

- [4] 杨红孺, 高洪元, 庞伟正. 基于离散粒子群优化算法的多用户检测器. 哈尔滨工业大学学报, 2005, 37 (9): 1303-1306.
- [5] 常青, 濮剑锋, 高洪元, 等. 基于改进的克隆选择算法的多用户检测技术. 航空学报, 2007, 28 (2): 391-396.
- [6] Gao Hongyuan, Cao Jinlong, Yu Xuemei. Multiuser Detection Based on the DNA Clonal Selection Algorithm. GCIS2010. 2010: 349-352.
- [7] Zhang J H, Huai J P, Xiao R Y, et al. Resource management in the next generation DS-CDMA cellular networks. IEEE Communications Magazine, 2004, 11(4):52-58.
- [8] Madhow U. MMSE interference suppression for timing acquisition and demodulation in direct-sequence CDMA systems. IEEE Trans. Commun, 1998, 46(8): 1065-1075.
- [9] 张贤达, 保铮. 通信信号处理. 北京: 国防工业出版社, 2000.
- [10] Tse D N C, Verdu S. Optimum asymptotic multiuser efficiency of randomly spread CDMA. IEEE Transactions on Information Theory, 2000, 46(7): 2718-2722.
- [11] 刁鸣, 高洪元, 贾宗圣, 等. 基于免疫量子算法的多用户检测技术研究. 哈尔滨工程大学学报, 2007, 28 (10): 1171-1175.
- [12] Manolakos E S. Hopfield neural network implementation of the optimal CDMA multiuser detector. IEEE Transactions on Neural Networks, 1996, 7(1): 131- 141.
- [13] 高洪元, 刁鸣, 赵忠凯. 基于免疫克隆量子算法的多用户检测器. 电子与信息学报, 2008, 30 (7): 1566-1570
- [14] 王磊, 潘进, 焦李成. 免疫算法. 电子学报, 2000, 28 (7): 74-78.
- [15] De Castro L N, Von Zuben F J. Learning and optimization using the clonal

- selection principle. IEEE Transactions on Evolutionary Computation, 2002, 6(3): 239-251.
- [16] Han K H, Kim J H. Genetic quantum algorithm and its application to combinatorial optimization problem. Proceedings of the 2000 IEEE Congress on Evolutionary Computation, 2000: 1354-1360 .
- [17] Ergun C, Hacıoglu K. Multiuser detection using a genetic algorithm in CDMA communications systems. IEEE Trans. Commun, 2000, 48(8): 1374-1383.
- [18] Kechriotis G, Manolakos E S. Hopfield neural network implementation in the optimal CDMA multiuser detector. IEEE Trans. Neural Networks, 1996, 7(1): 131-141.
- [19] 王永刚, 焦李成. 基于随机 Hopfield 神经网络的最优多用户检测器. 电子学报, 2004, 32 (10): 1630-1634.
- [20] 高洪元, 刁鸣, 贾宗圣. 基于神经网络量子算法的多用户检测器. 计算机工程, 2007, 33 (10): 196-198.
- [21] Kechriotis G I, Manolakos E S. Comparison of a neural network based receiver to the optimal and multistage CDMA multiuser detectors. Neural Networks for Signal Processing [1995] V. Proceedings of the 1995 IEEE Workshop. 1995: 613-622.
- [22] Zhang Jian, Peng Qicong, Shao Huaizong, Shao Tiange. Nonlinear multiuser detection based on SVM in fading channel with impulse noise. 2006 6th International Conference on ITS Telecommunications, 2006: 545-548.
- [23] 李昌彪, 夏克文, 宋建平, 等. 一种基于属性重要性的粗糙 RBF 神经网络. 控制与决策, 2006, 21 (7): 821-824.
- [24] 黄国宏, 熊志化, 邵惠鹤. 一种新的基于构造型神经网络分类算法. 计算机学报, 2005, 28 (9): 1519-1523.

- [25] Perotti J I, Tamarit F A, Cannas S A. A scale-free neural network for modeling neurogenesis. *Physica A: Statistical and Theoretical Physics*, 2006, 371(1): 71-75.
- [26] Sahoo G B, Ray C. Predicting flux decline in crossflow membranes using artificial neural networks and genetic algorithm. *Journal of Membrane Science*, 2006, 283(1/2):147-157.
- [27] 解光军. 量子神经计算及其模型研究. 合肥: 中国科学技术大学, 2002.
- [28] Toth G, Lent C S, Douglas P, et al. Quantum cellular neural networks. *Superlattices and Microstructures*, 1996, 20(4): 473-478.
- [29] Stuwart S L, Selvi S T. Precoder based collaborative blind multiuser detection of CDMA signals. *Wireless Personal Communications*, 2015, 83(1): 135-148.
- [30] 刁鸣, 高洪元, 马杰, 等. 应用神经网络粒子群算法的多用户检测. *电子科技大学学报*, 2008, 37 (2): 178-180.
- [31] 高洪元, 刁鸣. 基于克隆遗传量子算法的多用户检测. *哈尔滨工程大学学报*, 2008, 29 (1): 85-89.
- [32] 刁鸣, 邹丽. 模拟退火遗传禁忌搜索的多用户检测算法. *哈尔滨工程大学学报*, 2014, 35 (3): 373-377.
- [33] 刘洪武, 冯全源. 分集接收的 STBC-MC-CDMA 系统中基于 PSO 算法的多用户检测. *电子与信息学报*, 2009, 31 (1): 45-48.
- [34] Gao Hongyuan, Diao Ming. Multiuser Detection Using Immune Ant Colony Optimization. *2009 International Conference on Artificial Intelligence and Computational Intelligence*. 2009, (2): 109-113.
- [35] 高洪元, 柴晓辉, 刁鸣. 基于免疫进化规划的多用户检测技术研究. *智能系统学报*, 2007, 2 (2): 78-82.
- [36] 高洪元, 曹金龙. 量子蜂群算法及其在认知频谱分配中的应用. *中南大学学报 (自然科学版)*, 2012, 43 (12): 4743-4749.

- [37] 高洪元, 梁炎松, 刘丹丹. 新量子蜂群算法的鲁棒多用户检测. 计算机工程, 2016.
- [38] Zhao Zhijin, Peng Zhen, Zheng Shilian, et al. Cognitive radio spectrum allocation using evolutionary algorithms. IEEE Transactions on Wireless Communications, 2009, 8(9): 4421-4425.
- [39] 赵知劲, 彭振, 郑仕链, 等. 基于量子遗传算法的认知无线电频谱分配. 物理学报, 2009, 52 (2): 1358-1363.

第 5 章

基于量子群智能的认知 无线电决策引擎

认知无线电作为一种缓解频谱资源紧张的方法日益受到人们的关注，并在不断地发展壮大中。认知决策引擎作为实现 CR 的主要关键技术，吸引着越来越多的学者也投身于相关研究。认知无线电将人工智能技术与无线电技术相结合，其智能性主要依赖于认知决策引擎这一关键技术。认知决策引擎相当于 CR 的“大脑”，也是 CR 的核心技术之一，其首要的研究是如何完成决策过程。第一个需要探讨的问题是认知决策过程应该以分布式方式还是集中式方式来完成^[1]。智能性通常是基于分布式系统实现的，这对于实现认知网络是至关重要的，且认知无线电的决策过程是由不同设备协作完成的，因此分布式系统成为首要选择。认知决策引擎通过从经验和环境中学习知识并进行分析，然后智能地调整工作参数使其满足用户的通信需求，因此认知决策引擎可以被看作一个多目标优化问题。那么随之而来的第二个问题是如何选择决策算法。现有的决策算法有神经网络、遗传算法、蚁群算法等。选择一个符合并满足认知无线电系统要求的算法是十分重要的。

目前被广泛使用的认知决策引擎模型是由维吉尼亚工学院的无线通信中心（VC-CWT）于 2004 年设计的，其设计理念是基于测试平台的 CR 仿生决策模

型,采用线性加权的方式把多目标优化问题转换为单目标问题来求解,并利用GA来实现系统参数的优化^[2]。Newman等人提出了多载波系统中基于GA的认知决策引擎^[3],分析了增加变量参数对实现决策引擎技术的复杂度的影响^[4]。随后,又在此基础上,将前一代种群所得到的认知环感知结果作为当前代的初始信息,以加速算法收敛^[5]。但是由于GA受随机搜索机制的限制,收敛速度提升的幅度有限,也有易于陷入局部最优的缺点,因此并不能获得良好的通信性能。与此同时,国内一些学者也将经典智能算法用于解决认知引擎决策问题,如基于量子遗传算法^[6]、模拟退火遗传算法^[7]、二进制粒子群^[8]和量子粒子群算法^[9]等的认知无线电决策引擎方法。通过仿真分析可以看出,这些算法与GA相比,收敛性能有所提高,但是仍然存在收敛精度不高的缺点^[10]。且上述研究均考虑的是对加权后的单目标问题进行优化,并不能有效考虑到各个通信指标的要求。文献[11]和[12]将多目标算法应用于认知决策引擎,但是所提算法较为复杂,实时性较差,并不能普遍应用于认知决策引擎中。目前认知决策引擎的研究尚存在许多缺点和不足,研究空间还很大。利用智能算法与认知决策引擎相结合,设计出实时性较好且能兼顾各个通信指标要求的决策引擎是一个值得进行深入研究的方向。

随着用户对带宽及服务质量(QoS)需求的不断增长,无线通信技术正迅猛发展,由此带来的过量能量消耗是一个亟待解决的问题。绿色通信技术可以减少能量消耗和提高能量利用率,因此有望缓解这一难题^[13,14]。认知无线电具有感知周围无线通信环境和智能调整参数以适应环境的能力,被广泛认为具有实现绿色通信技术的潜能。随着人们对认知无线电的深入研究,旨在实现绿色通信技术的绿色认知无线电将掀起新的研究热潮。

本章主要介绍了基于膜量子蜂群算法的决策引擎和基于量子细菌觅食算法的绿色认知无线电参数调整。

5.1 认知无线电决策引擎模型和三种典型的决策引擎

本节介绍了认知决策引擎的智能优化模型，并给出了三种经典智能算法实现的认知无线电决策引擎。

5.1.1 智能计算的认知决策引擎

认知决策引擎是 CRS 中智能管理系统的核心，也是调整参数实现系统最优化的执行者。但什么是实现系统的“最优化”，怎样才能做到“最优化”呢？

最优化不仅仅是为了追求某些需求而自私地调整相应的参数，而是要在考虑多方面需求的情况下达到满足用户需求的性能。认知无线电系统是被认知决策过程所驱动的。通过认知决策引擎在 CRS 中重构必要的模拟、感知、学习及决策，进而实现无线网络的最优化。图 5.1.1 给出了通用认知无线电体系结构。从图中可以看出，认知决策引擎从用户域、外界环境、政策域及无线电自身等获得和网络需求相关的信息^[15]，并利用这些信息通过智能决策过程来实现无线电的最优化。无线电域的信息包括射频和可能影响系统性能的环境数据。政策引擎从政策域接收政策相关的信息，这些信息有助于认知决策引擎确定可行的解决方法。

认知无线电的目标是通过感知和学习等方式理解环境，并智能地调整参数以实现自身性能最优化且满足用户需求，资源的配置过程可通过对系统的可操作参数进行调整来实现。在 CRS 中，将系统参数分为两类：可写参数和可观测参数。其中可写参数指的是影响 CRS 的信道性能和无线电操作的参数，被认为是可操作参数。可观测参数通常代表无线电操作优化时衡量系统性能的最大化或最小化目标函数。其中部分可观测函数和可写参数如表 5.1.1 所示。

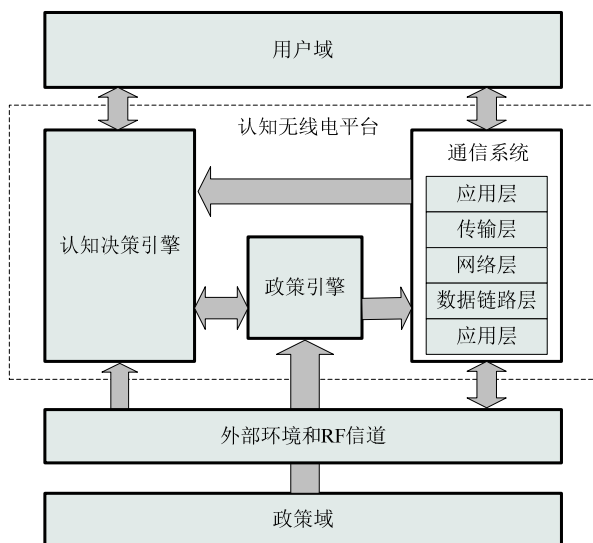


图 5.1.1 通用认知无线电体系结构

表 5.1.1 可观测参数与可写参数

可观测参数	可写参数
比特错误率	发射功率
信噪比	调制类型
数据速率	调制阶数
噪声功率	带宽
频谱效率	符号速率
干扰功率	载波频率

认知无线电决策引擎的主要作用是在频谱分配之后根据当前频谱环境，通过自适应调整控制参数（可写参数）来达到某些系统目标（可观测参数）^[16]。因此，决策的过程可简单概括为寻找某一参数组合以满足最优性能的过程。可以看出这是一个多目标优化的问题，根据上述过程描述可以建立数学模型为

$$\max \mathbf{Y} = \max[f_1(\mathbf{y}), f_2(\mathbf{y}), \dots, f_n(\mathbf{y})] \quad (5.1.1)$$

其中， n 为所需要优化的目标数， \mathbf{Y} 是目标函数向量， \mathbf{y} 是输入控制参数的向量， $f_i (i=1,2,\dots,n)$ 是归一化目标函数。

本书主要考虑三个优化目标，分别是最小化发射功率、最小化比特错误率和最大化数据速率。为了满足式 (5.1.1) 所示的数学模型，将待优化的 3 个目标写为归一化的函数表达式形式，分别为

$$f_1(\mathbf{y}) = 1 - \frac{\bar{p}}{p_{\max}} = 1 - \frac{1}{Np_{\max}} \sum_{i=1}^N y_i \quad (5.1.2)$$

$$f_2(\mathbf{y}) = 1 - \frac{\log_{10} 0.5}{\log_{10} \bar{p}_{be}} = 1 - \frac{\log_{10} 0.5}{\log_{10} \left(\frac{1}{N} \sum_{i=1}^N p_{be}^i \right)} \quad (5.1.3)$$

$$f_3(\mathbf{y}) = \frac{\frac{1}{N} \sum_{i=1}^N \log_2(y_{N+i}) - \log_2 M_{\min}}{\log_2 M_{\max} - \log_2 M_{\min}} = \frac{\frac{1}{N} \sum_{i=1}^N \log_2(M_i) - \log_2 M_{\min}}{\log_2 M_{\max} - \log_2 M_{\min}} \quad (5.1.4)$$

其中， $\mathbf{y} = (y_1, y_2, \dots, y_{2N})$ 为待优化的系统参数， $y_i (1 \leq i \leq N)$ 为第 i 个子载波的发射功率， $y_{N+i} = M_i (1 \leq i \leq N)$ 为第 i 个子载波的调制阶数，子载波总数为 N ，式中 \bar{p} 是 N 个子载波的平均发射功率， p_{\max} 表示最大发射功率； \bar{p}_{be} 表示平均误比特率； M_{\max} 表示最大的调制阶数， M_{\min} 表示最小的调制阶数。式 (5.1.3) 中的比特错误率 p_{be}^i 可根据不同的调制方式来计算。在高斯白噪声环境中，矩形 M_i -QAM 的比特错误率可由式 (5.1.5) 计算。

$$p_{be}^i = \frac{2 \left(1 - \frac{1}{\sqrt{M_i}} \right)}{\log_2(M_i)} \operatorname{erfc} \left(\sqrt{\frac{3 \log_2(M_i)}{2(M_i - 1)}} \times \frac{E_b^i}{n_0} \right) \quad (5.1.5)$$

在高斯白噪声环境中， M_i -PSK 的比特错误率可由式 (5.1.6) 计算。

$$p_{be}^i = \frac{1}{\log_2 M_i} \operatorname{erfc} \left(\sin \left(\frac{\pi}{M_i} \right) \sqrt{\log_2(M_i) \cdot \frac{E_b^i}{n_0}} \right) \quad (5.1.6)$$

其中， E_b^i 表示第 i 个子载波的比特能量，和此载波的发射功率 y_i 相关； n_0 表示高斯白噪声的功率谱密度。其他调制方式的误码率公式可参考相关文献。

对于此问题模型，应该利用多目标优化算法求得非支配解集，但现有研究为了简化计算，通常将此多目标优化问题通过设置权重以加权的方式转化为单目标优化问题，具体如式 (5.1.7) 所示。

$$f(y) = \sum_{i=1}^3 w_i f_i(y), w_i \geq 0, \sum_{i=1}^3 w_i = 1 \quad (5.1.7)$$

其中, w_i 表示目标函数 f_i 在决策过程中的权重, 权重数值的大小反映了此目标函数的重要程度。例如在数据传输的通信中, 最小化比特错误率将是关注的侧重点, 则将其相应目标函数赋予较大的权重值。具体权重可根据具体的通信需求来设置。上述数学模型建立完成, 认知决策引擎的决策过程可通过选择合适的智能决策算法来完成, 下一节将介绍几种基于智能计算的认知引擎决策方法。

5.1.2 基于智能计算的认知决策引擎方法

1. 基于遗传算法的认知决策引擎

遗传算法是基于生物进化过程中的自然选择理论的一种仿生进化算法, 其种群由一组代表潜在解的染色体群体所构成, 在种群中具有较大适应度值的染色体(个体)将被保存到子代, 并以较大的可能性通过不同的遗传操作产生更优秀的解, 即趋近于最优解^[17]。

在基于遗传算法的认知决策引擎中, 每一组系统参数可以表示为一个用单链二进制字符串编码的个体或染色体, 每一个二进制比特可看作一个个体或染色体的基因, 种群的进化是通过对基因进行操作而完成的^[18]。利用所构造的决策引擎单目标函数[如式(5.1.7)所示]作为衡量每个个体的适应度函数, 适应度值越大, 与其对应的个体越优秀。所以基于遗传算法的认知引擎的决策过程可以看作搜索适应度值最大个体的过程。遗传算法主要通过下述三种算子来完成算法的演进过程。

1) 选择算子

选择算子的主要操作是通过对个体进行适应度值评价, 挑选一些适应度值较大的个体进入下一代进行繁衍。挑选的个体过于集中会导致种群局部收敛, 挑选的个体过于发散会导致所求解远离全局最优解, 因此选择算子是种群演化过程中一个非常重要的算子。较为常用的选择算子为轮盘赌规则, 依据计算得

来的个体的适应度值占整个种群适应度值的比例，依概率选择进入下一代的个体。如图 5.1.2 所示为一个轮盘赌示意图，假设存在一个包含 6 个个体的种群，其中每个个体所占总体的比例在图中可清晰地表现出来。个体 1 在图中占有最大的比例，说明其对应的适应度值最大，也意味着进入下一代的可能性最大；相反，个体 2 被选择进入下一代的概率最小。

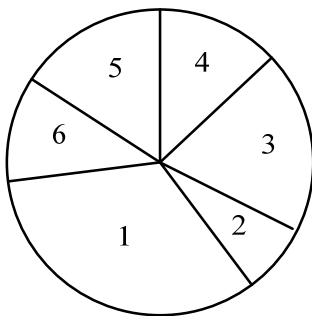


图 5.1.2 轮盘赌示意图

2) 交叉算子

交叉算子也可称为重组算子，它是 GA 演进过程中产生新个体的基本操作之一。与自然界进化相仿，交叉产生的个体具有两个父代个体的基因，即包含两个父代个体的遗传信息。在 GA 中，最简单且被广泛使用的是单点交叉。单点交叉的过程可描述为：对种群中的个体以两个为一组随机进行配对，对完成该过程后的每一组个体随机选择一个基因位作为交叉点，并依概率进行交叉重组。具体的交叉过程如图 5.1.3 所示。

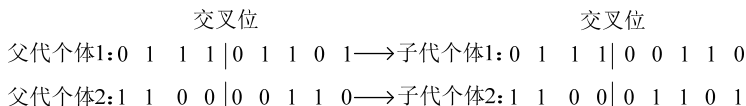
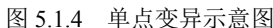


图 5.1.3 单点交叉示意图

3) 变异算子

在自然进化过程中，变异是一个随机的过程，染色体的某一基因被其他基因替代进而产生一个新的基因结构。在遗传算法中，变异是以小概率发生的事



—— 从《孟子》看《荀子》——

—— 每个人在根据本书 (1) 中方法不断修正自己的信念时, 本书只应阅读一次。

—— 臺灣林業地圖與林業地圖 —— 以臺灣林業發展與林業地圖為例

Figure 1. The relationship between the number of species and the number of individuals in the assemblages. The number of species is plotted against the number of individuals in the assemblages. The data points are shown as open circles. The solid line represents the expected relationship between the number of species and the number of individuals in the assemblages. The dashed line represents the observed relationship between the number of species and the number of individuals in the assemblages.

2. 基于量子遗传算法的认知决策引擎

QGA 是在 GA 的基础上引入量子计算理论而提出的, 其中量子计算具体内容包量子比特的概念及叠加态的表达方式。QGA 使用量子态的表达方法替代了现有的利用二进制、实数等其他表达符号进行编码的方式。其主要步骤包括量子编码、种群测量及种群更新等^[6]。

在 QGA 中, 种群 $Q(g)$ 包含 P 个由量子比特构成的染色体, P 为种群规模。在第 g 次迭代中, 一个包含 D 位量子比特的第 i 个染色体可以表示为

$$q_i^g = \begin{bmatrix} \alpha_{i1}^g, \alpha_{i2}^g, \dots, \alpha_{iD}^g \\ \beta_{i1}^g, \beta_{i2}^g, \dots, \beta_{iD}^g \end{bmatrix} \quad (5.1.8)$$

其中, $|\alpha_{ij}^g|^2 + |\beta_{ij}^g|^2 = 1$ ($i=1, 2, \dots, P$, $j=1, 2, \dots, D$); $|\alpha_{ij}^g|^2$ 表示第 i 个染色体第 j 个量子比特选择状态“0”的概率; $|\beta_{ij}^g|^2$ 表示第 i 个染色体第 j 个量子比特选择状态“1”的概率。

通过对种群 $Q(g)$ 中的每个个体进行测量, 可以获得 P 组由二进制解组成的 $P(g)$, 即 $P(g)$ 中每一个解都是由长度为 D 的二进制字符串构成的。在 $P(g)$ 中, 第 i 个染色体的第 j 位的取值由其所对应的 $|\alpha_{ij}^g|^2$ ($j=1, 2, \dots, D$) 来确定, 通过产生在 $[0, 1]$ 之间的随机数, 并将其与 $|\alpha_{ij}^g|^2$ 相比较, 如果小于 $|\alpha_{ij}^g|^2$, 则第 i 个染色体第 j 位取值为“0”, 否则为“1”。

对每个个体进行适应度评价实际上是对 $P(g)$ 中由二进制解构成的个体进行评价。初始化种群 $Q(g)$ 时, 将每个个体的各量子比特位均初始化为 $1/\sqrt{2}$, 使得在算法的初始搜索时每个量子位选择状态“0”和状态“1”的概率相等。

在 QGA 中使用量子旋转门 $U(\theta_{ij}^{g+1}) = \begin{bmatrix} \cos \theta_{ij}^{g+1} & -\sin \theta_{ij}^{g+1} \\ \sin \theta_{ij}^{g+1} & \cos \theta_{ij}^{g+1} \end{bmatrix}$ 来完成对每个量子比特的更新操作, 具体方法为

$$\begin{bmatrix} \alpha_{ij}^{g+1} \\ \beta_{ij}^{g+1} \end{bmatrix} = U(\theta_{ij}^{g+1}) \begin{bmatrix} \alpha_{ij}^g \\ \beta_{ij}^g \end{bmatrix} \quad (5.1.9)$$

其中, $\begin{bmatrix} \alpha_{ij}^{g+1} \\ \beta_{ij}^{g+1} \end{bmatrix}$ 为 $\begin{bmatrix} \alpha_{ij}^g \\ \beta_{ij}^g \end{bmatrix}$ 在使用 $U(\theta_{ij}^{g+1})$ 更新后的量子比特形式, θ_{ij}^{g+1} 为完成此过程的量子旋转角。通过确定恰当的量子旋转角, 使得每一个量子比特向着最优的方向演进。量子旋转角的确定方法为 $\theta_{ij}^{g+1} = \Delta\theta_{ij}^{g+1} \cdot s(\alpha_{ij}, \beta_{ij})$, 在此方程中, $\Delta\theta_{ij}^{g+1}$ 和 $s(\alpha_{ij}, \beta_{ij})$ 分别表示量子旋转角的旋转角度及旋转方向。在对每个个体进行适应度值计算后, 将所获的结果与当前迭代中所保留的最优秀二进制解所对应的适应度进行比较, 如果此值优于最优秀个体的适应度, 那么需要调整该个体所对应位置的量子比特, 使得 $\begin{bmatrix} \alpha_{ij}^g \\ \beta_{ij}^g \end{bmatrix}$ 向着有助于出现该二进制解的方向演进; 否则, 需要调整该量子比特使得 $\begin{bmatrix} \alpha_{ij}^g \\ \beta_{ij}^g \end{bmatrix}$ 向着有助于出现已保存的最优秀二进制解的方向演进。

将 QGA 用于解决认知决策引擎问题, 可概括为如下步骤^[6] :

Step 01 初始化种群 $Q(g)$, 设置迭代次数 $g = 0$ 。

Step 02 对种群 $Q(g)$ 中的每个个体进行一次测量, 进而获得由二进制解组成的 $P(g)$, 并将 $P(g)$ 中的每个个体映射为认知无线电系统参数。

Step 03 对 $P(g)$ 中的每个个体进行适应度函数 (即决策引擎模型的合成目标函数) 计算, 并将最优秀的二进制解保存。

Step 04 如果 g 未达到设置的最大迭代次数, 那么进入 **Step 05** 继续运行; 否则, 算法终止。

Step 05 更新所有个体的量子旋转角和量子比特, 可以获得新的量子种群 $Q(g+1)$ 。

Step 06 令 $g = g + 1$ 。对种群 $Q(g)$ 进行测量, 获得 $P(g)$, 并将 $P(g)$ 中的每个个体映射为认知无线电系统参数。

Step 07 对每个量子个体当前的二进制解进行适应度评价。

Step 08 进行保留最优二进制解操作, 并转至 **Step 04**。

3. 基于粒子群算法的认知决策引擎

粒子群算法自学者 Kenney 等人提出后, 便引起各界学者的关注, 因其易于实现、参数较少且优化效果良好, 被广泛应用在各工程领域中。

离散 PSO 是由连续 PSO 发展而来的, PSO 每个粒子所对应的位置被看作待优化问题的潜在解, 在求解决策引擎问题时, 每个解也可被映射为决策引擎中可操作的系统参数。PSO 利用每个粒子的最优位置信息和整个种群的最优位置信息进行位置更新和种群演化。假设应用于认知决策引擎的粒子总数为 P , 在 D 维空间内进行飞行搜索。在第 g 次迭代中, 第 i 个粒子在空间中的位置记作 $\mathbf{x}_i^g = [x_{i1}^g, x_{i2}^g, \dots, x_{iD}^g]$, 其所对应的速度可以表示为 $\mathbf{v}_i^g = [v_{i1}^g, v_{i2}^g, \dots, v_{iD}^g]$ 。每个粒子所在位置的优劣性可通过其对应的适应度值来进行评价, 适应度值越高, 该粒子所对应的位置越优秀。到第 g 次迭代为止, 第 i 个粒子所搜索的最优秀的位置表示为 $\mathbf{p}_i^g = [p_{i1}^g, p_{i2}^g, \dots, p_{iD}^g]$, 整个粒子群所搜索的最优秀的位置表示为 $\mathbf{p}_b^g = [p_{b1}^g, p_{b2}^g, \dots, p_{bD}^g]$, 其中下标 b 是最优秀粒子的索引值。根据上述介绍, 粒子 i 的速度和其对应位置的更新表达式如下

$$\mathbf{v}_{id}^{g+1} = \mathbf{v}_{id}^g + c_1 r_1 (\mathbf{p}_{id}^g - \mathbf{x}_{id}^g) + c_2 r_2 (\mathbf{p}_{bd}^g - \mathbf{x}_{id}^g) \quad (5.1.10)$$

$$\text{sigmiod}(\mathbf{v}_{id}^{g+1}) = \frac{1}{1 + \exp(-\mathbf{v}_{id}^{g+1})} \quad (5.1.11)$$

$$\mathbf{x}_{id}^{g+1} = \begin{cases} 1, & \rho_{id}^{g+1} < \text{sigmiod}(\mathbf{v}_{id}^{g+1}) \\ 0, & \rho_{id}^{g+1} \geq \text{sigmiod}(\mathbf{v}_{id}^{g+1}) \end{cases} \quad (5.1.12)$$

其中, $i=1, 2, \dots, P$, $d=1, 2, \dots, D$; c_1 和 c_2 是两个常数, 表示学习因子, 一般取 2; r_1 、 r_2 和 ρ_{id}^{g+1} 为在 $[0, 1]$ 之间均匀分布的随机数; $\mathbf{v}_{id}^{g+1} \in [-v_{\max}, v_{\max}]$, v_{\max} 为粒子速度最大值, 通过其限制速度的变化范围幅值, 通常将其设为 4。

将 PSO 应用于解决认知决策引擎的流程如下^[8]:

Step 01 设 $g = 0$, 在可行空间内以随机的方式初始化粒子的位置与速度, 同样也初始化粒子 i ($i=1, 2, \dots, P$) 所搜索的最优秀的位置 $\mathbf{p}_i^g = [p_{i1}^g, p_{i2}^g, \dots, p_{iD}^g]$ 和整个种群的最优秀位置 $\mathbf{p}_b^g = [p_{b1}^g, p_{b2}^g, \dots, p_{bD}^g]$ 。

Step 02 将每个粒子位置映射为认知无线电系统参数，并按照式 (5.1.7) 计算粒子的适应度值。

Step 03 按照式 (5.1.10) 来更新各粒子速度，若其值在 $[-v_{\max}, v_{\max}]$ 外，便将其设置为最近的边界值上。

Step 04 按照式 (5.1.11) 和式 (5.1.12) 来更新每个粒子的位置。

Step 05 令 $g = g + 1$ ，对每个粒子进行适应度值计算。如果粒子 i 所对应位置的适应度值高于 p_i^{g-1} 的适应度值，那么 p_i^g 将被赋值为当前位置，否则 p_i^g 取值为 p_i^{g-1} ；如果 p_i^g 的适应度值同样高于 p_b^{g-1} 所对应的适应度值，那么 p_b^g 和 p_b^{g-1} 将被赋值为 p_i^g ，否则 p_b^g 将被赋值为 p_b^{g-1} 。

Step 06 如果 g 达到预先设置的最大迭代次数，则粒子群算法终止，并获得最优秀的粒子；否则，返回 **Step 03**。

5.2 单目标膜量子蜂群算法及其在决策引擎上的应用^[20]

5.1 节介绍了利用已有的经典算法解决认知无线电决策引擎问题的方法，但是这些经典智能算法受收敛精度与收敛速度的制约，在求解优化问题时并不能获得较好的收敛性能。因此，本节对其进行深入研究，并有针对性地进行改进与提升，创新设计了膜量子蜂群算法，利用膜计算和量子计算理论与人工蜂群算法理论相结合，有效地提升了算法的收敛能力。同时可以综合考虑各性能指标要求，从而提高整个系统的性能。仿真结果表明，基于膜量子蜂群算法的认知决策引擎方法相对于经典方法均能获得更优秀的性能。

5.2.1 膜结构简介

膜系统是一种受生物细胞的结构与功能启发而得到的分布式并行计算模型，一个膜系统也可称为 P 系统。如图 5.2.1 所示为一个表层膜内包含 6 个膜的简

单膜结构，可以记作 $[_0[_1[_3]_3]_1[_2[_4]_4[_5]_5]_2[_6]_6]_0$ 。膜系统是由一系列分层排列的膜构成的，所有的非表层膜均被嵌在表层膜中^[21]。基础膜为内部不包含其他膜的膜，每个膜可被看作一个区间，每个区间又可以组成膜结构的不同部分，也可包含多个目标集、交流规则和转换规则，计算结果可由表层膜输出。

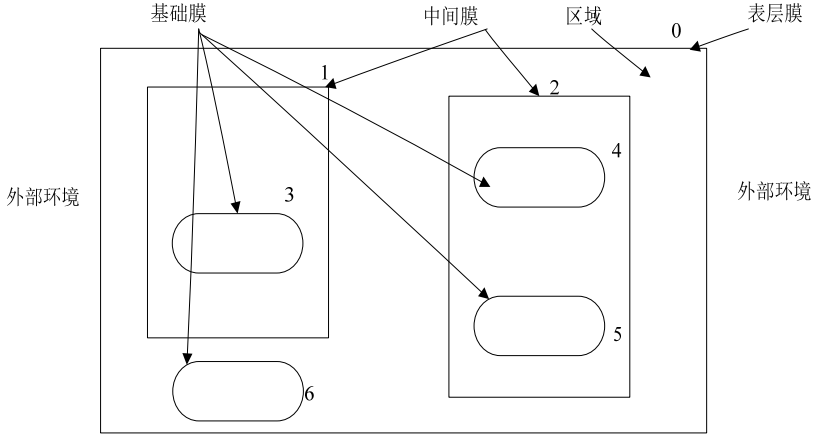


图 5.2.1 膜结构示意图

参考文献[21]，设计膜量子蜂群优化的膜结构如下

$$\Pi = (V, T, \mu, u_0, u_1, \dots, u_m, R_0, R_1, \dots, R_m, i_0) \quad (5.2.1)$$

其中， V 表示一个字母表，它内部的元素可以被称为对象； $T \subseteq V$ 表示输出字母表； μ 是一个由 $m+1$ 个膜以分层嵌入方式构成的膜结构，其中每个膜使用标号集 $\{0, 1, 2, \dots, m\}$ 中的元素表示其范围， $m+1$ 为膜结构的度数； $u_k, 0 \leq k \leq m$ ，表示与 μ 的第 k 个区域相互关联的多重集； $R_k, 0 \leq k \leq m$ ，是与 μ 的第 k 个区域相互关联的进化规则的有限集； i_0 为0和 m 之间的整数，表示 Π 的输出膜序号，选择标号为0的表层膜作为输出膜。

5.2.2 膜量子蜂群优化算法

膜量子蜂群优化（MQBCO）通过模拟膜结构，并利用P系统的一些要素，进而完成并行搜索机制。设每个基础膜是一个可以任意变化形状的透明弹性膜，

因此蜜蜂可在其中到达搜索空间的任何位置。膜结构中字母表的对象为量子比特和二进制比特 $\{0,1\}$ ，分别组成蜜蜂的量子位置和位置。膜系统中的演进规则用于演进所设计的量子蜂群算法。

设蜂群中共有 h 只蜜蜂，每只蜜蜂的量子位置由一串量子比特构成。根据量子计算基本理论，蜂群中第 i 只蜜蜂的量子位置表示为

$$\mathbf{v}_i = \begin{bmatrix} \alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iD} \\ \beta_{i1}, \beta_{i2}, \dots, \beta_{iD} \end{bmatrix} \quad (5.2.2)$$

其中， $|\alpha_{id}|^2 + |\beta_{id}|^2 = 1$ ($d=1,2,\dots,D$)。为便于演化公式设计，将 α_{id} 和 β_{id} 定义为 $0 \leq \alpha_{id} \leq 1$, $0 \leq \beta_{id} \leq 1$ 。膜量子蜂群优化通过对蜂群中每只蜜蜂量子位置进行更新完成算法的演进过程。 θ_{id}^{t+1} 为第 $t+1$ 次迭代的量子旋转角，每个量子

比特 $\mathbf{v}_{id}^t = [\alpha_{id}^t, \beta_{id}^t]^T$ 可以用量子旋转门 $U(\theta_{id}^{t+1}) = \begin{bmatrix} \cos \theta_{id}^{t+1} & -\sin \theta_{id}^{t+1} \\ \sin \theta_{id}^{t+1} & \cos \theta_{id}^{t+1} \end{bmatrix}$ 或量子非门

$G = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ 进行更新。

在 MQBCO 中，每只蜜蜂对应一个食物源位置，每个食物源位置用 D 维二进制比特组成。蜜蜂食物源位置的优劣性由其所对应的蜜源含量值所决定，蜜源含量越高，该位置越优秀。对于最大值优化问题，蜜源函数值越大代表蜜源含量高；对于最小值优化问题，蜜源函数值越小代表蜜源含量越高。第 i 只蜜蜂搜索到的食物源位置为 $\mathbf{x}_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t]$ ，这也是优化问题的一个潜在解。第 i 只蜜蜂在第 t 次迭代过程中搜索到的蜜源含量最高的位置记作局部最优位置 $\mathbf{p}_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t]$ ($i=1,2,\dots,h$)。到现在为止整个量子蜂群所搜索到的蜜源含量最高的位置，记作全局最优位置 $\mathbf{p}_g^t = [p_{g1}^t, p_{g2}^t, \dots, p_{gD}^t]$ 。

每个细胞膜内的蜂群由工蜂和观察蜂构成，数量各占一半。每只工蜂或者观察蜂都对应一个量子位置，对量子位置进行测量可以得到相应的食物源位置。每只蜜蜂根据以往的觅食经验和整个蜂群的信息交流，在所有的可能解区间搜索最优秀的食物源位置。

在 MQBCO 中， R_k ($1 \leq k \leq m$) 是第 k 个膜的演进规则。把所有的工蜂和观

观察蜂平均分配到 m 个基础膜中，每个基础膜内都有 $0.5h/m$ 只工蜂和 $0.5h/m$ 只观察蜂。

第 i 只工蜂的量子旋转角通过局部最优位置 \mathbf{p}_i^t 和全局最优位置 \mathbf{p}_g^t 来确定及更新。在第 $k(1 \leq k \leq m)$ 个基础膜内，每只工蜂的量子旋转角及量子位置可分别使用式 (5.2.3) 和式 (5.2.4) 进行更新。

$$\theta_{id}^{t+1} = e_1(p_{id}^t - x_{id}^t) + e_2(p_{gd}^t - x_{id}^t) \quad (5.2.3)$$

$$\mathbf{v}_{id}^{t+1} = \begin{cases} \mathbf{G}\mathbf{v}_{id}^t, & \theta_{id}^{t+1}=0 \text{ 且 } \gamma_{id}^{t+1} < c_1 \\ \text{abs}[U(\theta_{id}^{t+1})\mathbf{v}_{id}^t], & \text{其他} \end{cases} \quad (5.2.4)$$

其中， $i = (k-1)h/m+1, (k-1)h/m+2, \dots, (k-0.5)h/m$ ， $d=1, 2, \dots, D$ ， $k=1, 2, \dots, m$ ； e_1 和 e_2 是两个常数，分别表示量子旋转角受工蜂局部最优位置和该膜中全局最优位置影响的权重值； γ_{id}^{t+1} 为 $[0,1]$ 之间均匀分布的随机数； c_1 是旋转角为 0 时量子位的翻转概率，其值可选 $[0,1/D]$ 之间的常数。

观察蜂通过观察工蜂的采蜜过程来挑选工蜂，并利用挑选工蜂的信息来更新观察蜂的位置信息。在第 $k(1 \leq k \leq m)$ 个基础膜内，每只观察蜂的量子旋转角及量子位置通过式 (5.2.5) 和式 (5.2.6) 进行更新。

$$\theta_{id}^{t+1} = e_3(p_{id}^t - x_{id}^t) + e_4(p_{jd}^t - x_{id}^t) + e_5(g_{kd}^t - x_{id}^t) \quad (5.2.5)$$

$$\mathbf{v}_{id}^{t+1} = \begin{cases} \mathbf{G}\mathbf{v}_{id}^t, & \theta_{id}^{t+1}=0 \text{ 且 } \gamma_{id}^{t+1} < c_2 \\ \text{abs}[U(\theta_{id}^{t+1})\mathbf{v}_{id}^t], & \text{其他} \end{cases} \quad (5.2.6)$$

其中， $i = (k-0.5)h/m+1, (k-0.5)h/m+2, \dots, kh/m$ ； $\mathbf{g}_k^t = [g_{k1}^t, g_{k2}^t, \dots, g_{kD}^t]$ 是第 k 个膜在第 t 代所发现的蜜源含量最高的位置，记作全局最优位置；常数 e_3 、 e_4 和 e_5 分别代表量子旋转角受观察蜂的局部极值、所选择工蜂的局部极值和该膜中的全局极值所影响的权重值； $c_2 \in [0,1/D]$ 代表量子旋转角等于 0 时量子位的翻转概率； γ_{id}^{t+1} 为 $[0,1]$ 之间均匀分布的随机数。其中 $\mathbf{p}_j^t = [p_{j1}^t, p_{j2}^t, \dots, p_{jD}^t]$ 为第 k 个膜内第 j 只工蜂的局部最优位置，是由此膜内第 $i(i = (k-0.5)h/m+1, (k-0.5)h/m+2, \dots, kh/m)$ 只观察蜂根据轮盘赌规则依概率进行选择的。 $f(\mathbf{p}_j^t)$ 代表第 j 只工蜂在第 k 个基础膜内局部最优位置的适应度值，构造的适应度函

数需大于 0 且值越大其对应位置越优秀。对于最大值优化问题，工蜂局部最优位置所对应的适应度值越大，它被观察蜂挑选的概率越大，则第 $j(j = (k-1)h/m+1, (k-1)h/m+2, \dots, (k-0.5)h/m)$ 只工蜂被选择的概率为

$$q_j^{t+1}(k) = \frac{f(p_j^t)}{\sum_{l=(k-1)h/m+1}^{(k-0.5)h/m} f(p_l^t)} \quad (5.2.7)$$

对第 i 只蜜蜂的量子位置的各量子位进行测量并获得食物源位置的方法如式 (5.2.8) 所示。

$$x_{id}^{t+1} = \begin{cases} 1, & \rho_{id}^{t+1} > (\alpha_{id}^{t+1})^2 \\ 0, & \rho_{id}^{t+1} \leq (\alpha_{id}^{t+1})^2 \end{cases} \quad (5.2.8)$$

其中， $d=1,2,\dots,D$ ； $\rho_{id}^{t+1} \in [0,1]$ 是一个均匀分布的随机数； $(\alpha_{id}^{t+1})^2$ 描述了在第 $t+1$ 代时量子位 v_{id}^{t+1} 选择状态“0”的概率。

基于量子蜂群的膜框架构成如下：

(1) 有 m 个区域，均存在于标号为 0 的表层膜中，将此膜结构记作 $[0]_1[1]_2[2]_3 \dots [m]_m$ 。

(2) 字母表是由可能的量子位置和二进制位置向量构成的集合。

(3) 输出端字母表 T 是由二进制位置向量构成的集合。

(4) 多重集 $u_0, u_1, u_2, \dots, u_m$ 是由多个量子位置集合构成的，可记作 $u_1 = \{v_1, v_2, \dots, v_{n_1}\}$ ， $u_2 = \{v_{n_1+1}, v_{n_1+2}, \dots, v_{n_1+n_2}\}$ ， \dots ， $u_m = \{v_{n_1+n_2+\dots+n_{m-1}+1}, v_{n_1+n_2+\dots+n_{m-1}+2}, \dots, v_{n_1+n_2+\dots+n_{m-1}+n_m}\}$ ，其中 $v_i(1 \leq i \leq h)$ 为第 i 只蜜蜂的量子位置； $n_j = h/m$ ($1 \leq j \leq m$) 是 u_j 中的蜜蜂数； $\sum_{j=0}^m n_j = h$ ， h 为量子蜂群中的蜜蜂总数。

(5) 膜结构的规则包括演化规则、测量规则与通信规则。

① 演化规则：在每个基础膜内的蜜蜂均利用量子旋转门去更新其量子位置 [具体演进参考式 (5.2.3) ~ 式 (5.2.6)]。

② 测量规则：通过对量子位置的测量可以得到蜜蜂的食物源位置 [见式 (5.2.8)]。

③ 通信规则：表层膜和基础膜进行信息交流，每个基础膜将当前膜内的全局最优位置及其对应的蜜源含量值（目标函数值）信息传送到表层膜，然后由表层膜把全局最优位置信息和蜜源含量值信息返回到各个基础膜，用于指导下一次迭代。

在 MQBCO 算法的演进过程中，基础膜 k ($1 \leq k \leq m$) 中第 i ($\frac{hk}{m} + 1 \leq i \leq \frac{h(k+1)}{m}$) 只蜜蜂的局部最优位置 p_i^t 通过下式进行更新

$$p_i^{t+1} = \begin{cases} x_i^{t+1}, & f(p_i^t) < f(x_i^{t+1}) \\ p_i^t, & f(p_i^t) \leq f(x_i^{t+1}) \end{cases} \quad (5.2.9)$$

其中， $f()$ 代表适应度函数。第 k 个基础膜中全局最优位置更新公式为

$$g_k^{t+1} = \arg \max_{p_i^{t+1}} \{f(p_i^{t+1})\}, \frac{hk}{m} + 1 \leq i \leq \frac{h(k+1)}{m} \quad (5.2.10)$$

整个蜂群的全局最优位置 p_g^t 利用各基础膜的全局最优位置来更新，更新公式为

$$p_g^{t+1} = \arg \max_{g_k^{t+1}} \{f(g_k^{t+1})\}, 1 \leq k \leq m \quad (5.2.11)$$

5.2.3 膜量子蜂群优化算法的性能测试

利用基准函数的最小值优化来评价所提出的 MQBCO 算法。设获得测试函数数值最小（等同于花蜜含量最高）的位置为最优位置，设蜜源函数与测试函数相同，则构造的适应度函数是蜜源函数与 10^{-10} 和的倒数。为了验证所提算法的高效性，与其他三种经典智能算法进行对比，包括 QGA、PSO 及量子蜂群算法（QBCO）。这 4 种智能算法的一些参数设置相同：种群中个体数目为 20，最大迭代次数设为 1000 代。QGA 的参数设置：量子旋转角从初始代的 0.1π 线性递减至最后一代的 0.005π [6]。PSO 的参数设置：两个学习因子被设置相等且为 2， $v_{\max} = 4$ [8]。对于 QBCO，参数设置参见文献[23]。对于 MQBCO，参数设置情况如下：基础膜数 $m = 2$ ($n_1 = 10, n_2 = 10$)，则相应的膜结构被设置为

$[_0l_1]_1[_2l_2]_0$ 。每个基础膜中算法的演化规则设置 R_1 为 $e_1 = 0.06$, $e_2 = 0.03$, $e_3 = 0.06$, $e_4 = 0.015$, $e_5 = 0.005$, $c_1 = c_2 = 0.01/D$; R_2 为 $e_1 = 0.06$, $e_2 = 0.03$, $e_3 = 0.06\pi$, $e_4 = 0.015\pi$, $e_5 = 0.005\pi$, $c_1 = c_2 = 0.01/D$ 。

所选取的基准函数分别为 Griewank 函数、Rosenbrock 函数、Schwefel 函数和 Rastrigin 函数, 其公式分别为

$$F_1(\mathbf{y}) = \frac{1}{4000} \left(\sum_{i=1}^q (y_i - 100)^2 \right) - \left(\prod_{i=1}^q \cos \left(\frac{y_i - 100}{\sqrt{i}} \right) \right) + 1, \quad (-600 \leq y_i \leq 600, i=1, 2, \dots, q) \quad (5.2.12)$$

$$F_2(\mathbf{y}) = \sum_{i=1}^{q-1} 100(y_{i+1} - y_i)^2 + (y_i - 1)^2, \quad (-50 \leq y_i \leq 50, i=1, 2, \dots, q) \quad (5.2.13)$$

$$F_3(\mathbf{y}) = 2 \times 418.9829 - \sum_{i=1}^q y_i \sin(\sqrt{|y_i|}), \quad (-500 \leq y_i \leq 500, i=1, 2, \dots, q) \quad (5.2.14)$$

$$F_4(\mathbf{y}) = \sum_{i=1}^q (y_i^2 - 10 \cos(2\pi y_i) + 10), \quad (-5.12 \leq y_i \leq 5.12, i=1, 2, \dots, q) \quad (5.2.15)$$

在下述仿真过程中, 采用二进制编码, 每个变量由 25 位二进制比特构成。对于所有测试函数, 参数 q 设置为 2, 所有在实验中获得的数据是 200 次仿真结果的平均值。平均最优目标函数值和迭代次数关系曲线如图 5.2.2~图 5.2.5 所示。

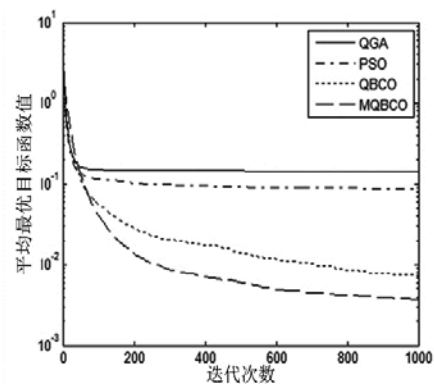


图 5.2.2 Griewank 函数的收敛曲线

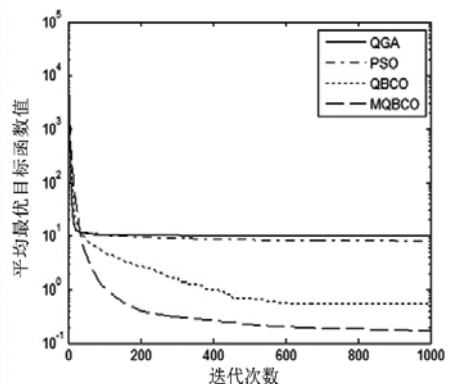


图 5.2.3 Rosenbrock 函数的收敛曲线

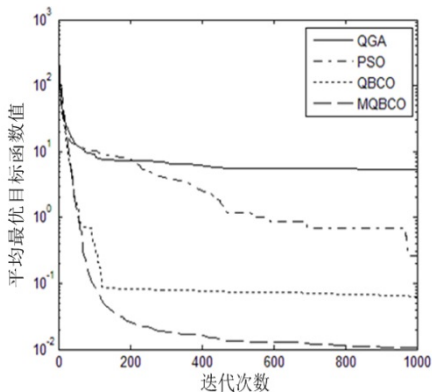


图 5.2.4 Schwefel 函数的收敛曲线

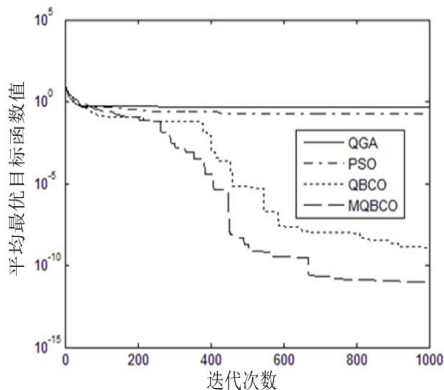


图 5.2.5 Rastrigin 函数的收敛曲线

由图 5.2.2 可知, 尽管经典算法 (QGA、PSO 和 QBCO) 的收敛速率较快, 但是这些算法均陷入局部最优值。图 5.2.2 也显示 MQBCO 有潜力获得更小的目标函数值, 也就是获得更精确的收敛精度。图 5.2.3 所示为 4 种算法对 Rosenbrock 函数的测试结果, 对于很多智能算法来说, 这是一个很难优化的函数。由于很难收敛到此函数的全局最优值, 因此该函数常被用来测试算法的性能。由图 5.2.3 可知, 所提出的 MQBCO 算法的收敛性能超过 QBCO、QGA 和 PSO。

Schwefel 函数是一个经典的很难被优化的函数, 含有多个峰值及谷值, 算法在对其进行测试时易陷入局部最小值。由图 5.2.4 可知, QGA 具有较快的收敛速度, 但是收敛精度不高; 而 PSO 能获得较好的收敛值, 但是收敛所需时间较长。而所设计的 MQBCO 无论在收敛精度还是收敛速度上均超过 QGA、PSO 和 QBCO。由图 5.2.5 可以看出, QGA 和 PSO 具有相似的性能, 均易陷入局部极值; 而所提出的 MQBCO 算法克服了 QBCO 局部收敛的缺点, 能够获得更好的收敛精度。

5.2.4 基于膜量子蜂群算法的认知无线电决策引擎

在 MQBCO 算法中, 将每只蜜蜂量子位置的各量子位初始化为 $1/\sqrt{2}$ 。

MQBCO 的初始位置集合是通过在解空间中随机初始化而得到的,这是由于对初始量子位进行测量获得二进制 $\{0,1\}$ 的概率是相等的,在此也可用随机初始化替代测量操作。蜜源函数(目标函数)和适应度函数被设为由加权方式转化而来的单目标认知决策引擎函数,对于最大值优化问题,蜜源含量值最大的位置为最优位置。认知无线电的决策过程可以被看作实现目标函数最大化的过程。

基于 MQBCO 的认知决策引擎优化方法的具体流程可概括如下:

Step 01 确定膜结构。初始化种群,具体包括二进制位置和量子位置,并将所有蜜蜂平均分配给各个基础膜。

Step 02 将每只蜜蜂的二进制位置映射成认知无线电系统参数。

Step 03 计算每只蜜蜂的花蜜含量,即计算其目标函数值,并初始化每只蜜蜂的局部最优位置、每个基础膜的全局最优位置和蜂群中最优秀的位置。

Step 04 通过演化规则与测量规则来更新每只蜜蜂的量子位置及其对应的食物源位置。

Step 05 将各蜜蜂的二进制位置映射成系统参数,根据蜜源函数计算其花蜜含量值。

Step 06 利用通信规则更新每只蜜蜂的局部最优位置、每个基础膜的全局最优位置和蜂群中最优秀的位置。

Step 07 如果达到所设置的最大迭代次数,则 MQBCO 终止并从表层膜输出全局最优解;否则,返回**Step 04**。

5.2.5 决策引擎试验仿真

将膜量子蜂群算法(MQBCO)运用于认知决策引擎中,分别与基于遗传算法(GA)、量子遗传算法(QGA)^[6]和粒子群算法(PSO)^[8]的认知决策引擎方法进行比较。仿真于包含 32 个子载波的 OFDM 系统,使用一个 $[0,1]$ 间的随

机数来模拟每个子载波信道的衰落。发射功率为 0.1-256.1mW。可选调制方式为 4-PSK、64-QAM、16-QAM 和 32-QAM，符号速率为 0.125Msps，信道噪声为加性高斯白噪声，其功率谱密度为 $1.4 \times 10^{-8} \text{ mW/Hz}$ 。这 4 种算法的初始种群、规模及最大迭代次数设置相同，种群规模为 20，终止迭代次数为 800。GA 的参数设置：交叉概率为 0.6，变异概率为 0.005。其他算法的参数设置与 5.2.3 节相同。每个量子蜜蜂的位置包括 256 位二进制比特位，其中每 8 位二进制比特对应一个子载波的参数信息（6 位二进制比特用于功率信息编码，2 位二进制比特用于对调制方式编码）。试验结果为 200 次试验的平均值。

考虑三种通信模式，模式 1 用于低功耗通信模式，模式 2 用于紧急通信模式，模式 3 用于多媒体通信模式。根据不同的方式进行权重设置，可以构建三种目标函数。具体的权重设置如表 5.2.1 所示。图 5.2.6~图 5.2.8 给出了三种通信模式下的目标函数值收敛曲线。

表 5.2.1 不同通信模式下的权重

权 重	模式 1	模式 2	模式 3
w_1	0.8	0.05	0.05
w_2	0.15	0.8	0.15
w_3	0.05	0.15	0.8

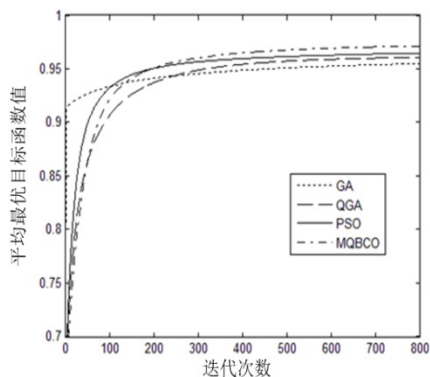


图 5.2.6 低功耗模式的性能收敛曲线

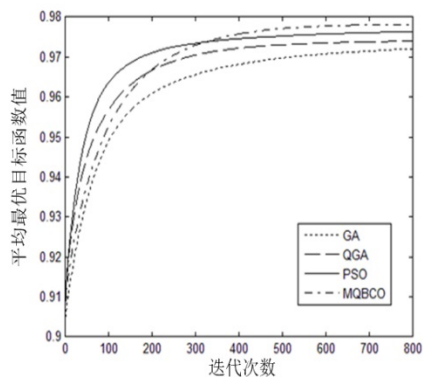


图 5.2.7 紧急模式的性能收敛曲线

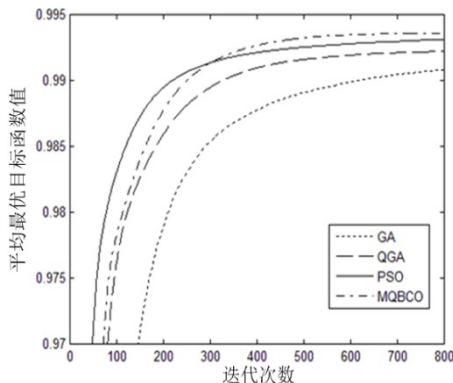


图 5.2.8 多媒体通信模式的收敛曲线

从图 5.2.6~图 5.2.8 可以看到 GA、PSO、QGA 和 MQBCO 4 种算法获得的收敛速度及收敛精度情况。低功耗通信模式下 4 种算法的收敛性能相近，但是所提出的 MQBCO 收敛精度要高于其他三种算法。在紧急通信模式和多媒体通信模式中，所提出的 MQBCO 所获得的收敛精度随着迭代次数增加而显著提升，并且与其他三种算法相比，MQBCO 仅需要较少的迭代次数便能达到较高的收敛精度，满足了认知决策引擎的实时性要求。仿真结果是 200 次试验的平均，再次说明了 MQBCO 的稳定性和高效性。

图 5.2.9~图 5.2.11 给出了三种通信模式下基于 MQBCO 的认知引擎对每个子载波的比特错误率、发射功率和通过量的参数调整结果。

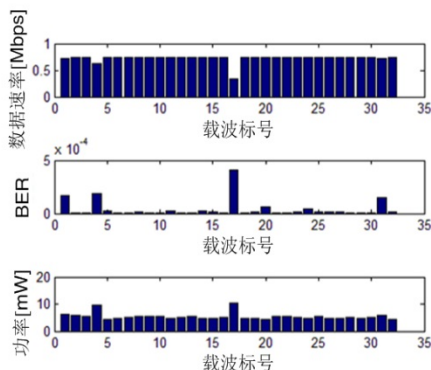


图 5.2.9 MQBCO 在低功耗通信模式下的参数调整结果

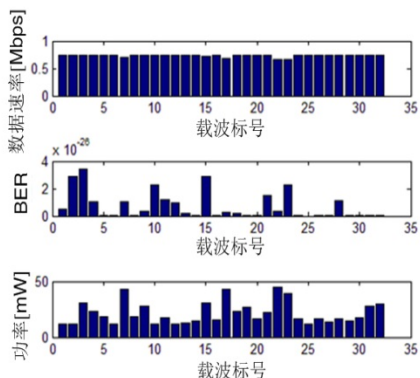


图 5.2.10 MQBCO 在紧急通信模式下的参数调整结果

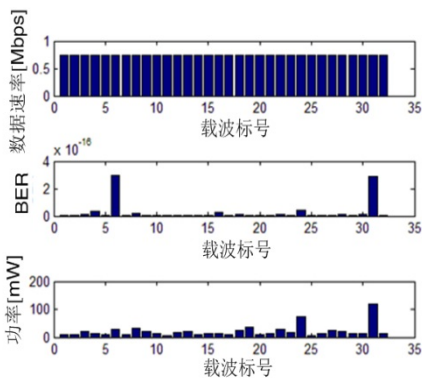


图 5.2.11 MQBCO 在多媒体通信模式下的参数调整结果

从图 5.2.9~图 5.2.11 所示的仿真结果可以看出,在低功耗通信模式下,所设计的基于 MQBCO 的认知决策引擎优化方法能满足系统的低功耗要求,即系统的能量消耗是高效的。在紧急通信模式下,在考虑用户发射功率和通过量需求的情况下,比特错误率被控制在较低的范围。在多媒体通信模式下,整个认知无线电系统的通过量得到提升。通过以上仿真试验,所提出的基于 MQBCO 的认知决策引擎参数优化方法的优越性得以验证。

5.3 基于量子细菌觅食算法的绿色认知无线电参数调整

近年来有很多学者致力于研究实现绿色认知无线电,其利用认知无线电相关关键技术研究了不同的方法来实现绿色通信技术。文献[24]提出了一种在输入反馈限制条件下实现绿色认知无线电机机会式频谱接入的方法。文献[25]研究了旨在最大化能量效率的认知和非认知功率发射机,并对如何配置认知无线电来提高异构网络的感知能力方面提出了深刻的见解。文献[26]研究了在协作认知无线电下行系统中系统性能和能量高效的均衡问题。文献[27]提出了一种适用于下行绿色认知无线电系统网络的功率迭代方法。上述研究在一定程度上实

现了绿色认知无线电技术，但是并没有考虑用户的服务质量需求。为此，文献[28]提出了一种基于粒子群算法的绿色认知无线电参数调整方法，该方法在考虑了用户的比特错误率和数据速率要求的同时降低了发射功率，从而实现能量高效的绿色认知无线电。但是此方法受粒子群算法局部收敛等缺点的限制，并不能获得所设计数学模型的最优解，因此，针对此模型设计一种更为智能高效的算法是十分必要的。

本节研究的重点是绿色认知无线电参数调整问题，该问题是认知决策引擎技术的延伸。研究的主要目的是在保证用户服务质量的条件下，通过调整参数来实现绿色通信，而不是盲目地降低发射功率。因此提出了一种新的绿色认知无线电参数调整方法，即基于量子细菌觅食算法的绿色认知无线电参数调整方法。与现有研究相比，所提出的方法能够很好地考虑到用户的 QoS 需求，并实现低功耗通信。

5.3.1 绿色认知无线电参数调整模型

绿色认知无线电的主要目标是实现能量的高效利用。但是在实际通信中，用户的通信需求及服务质量都应该被保证，然而这些需求也是随着用户的地理位置和通信环境等因素变化的，因此需要实时地调整参数以实现在保证用户服务质量的前提下最小化能量消耗，进而实现绿色认知无线电。

为满足上述要求，在本节中将给出绿色认知无线电参数调整的一个数学模型，所给出的问题模型也是基于一个多载波的系统。待优化调整的认知无线电系统参数包括发射功率、调制类型及调制阶数。本文中保证用户的服务质量指的是保证用户在通信中的比特错误率和数据速率可以达到目标值。因此，所设计的问题模型可以被描述为如何调整发射功率、调制类型及调制阶数来实现在保证用户服务质量的前提下，实现能量消耗的最小化。由此可以推断，这个过程可以看作一个有约束条件的参数优化问题，将其建立模型为

$$\min \sum_{k=1}^N p_k, \text{st. } BER_k \leq BER_{tar}, DR_k \geq DR_{tar} \quad (5.3.1)$$

其中, N 表示子载波的数目, p_k 是第 k 个子载波的发射功率, BER_{tar} 是用户的目标比特错误率, BER_k 是第 k 个子载波的比特错误率, DR_{tar} 是用户的目标数据速率, DR_k 是第 k 个子载波的数据速率。

为便于计算, 将上述有约束优化问题转化为无约束的参数优化问题, 可建立如下问题模型:

$$\max f \quad (5.3.2)$$

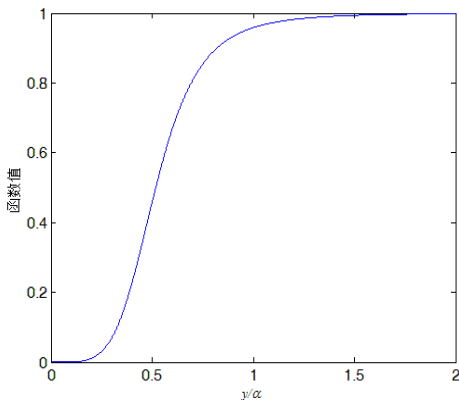
其中,

$$f = (Np_{\max} + 1 - \sum_{k=1}^N p_k) \times \bar{f} \left(\left(\frac{1}{N} \sum_{k=1}^N BER_k \right)^{-1}, BER_{tar}^{-1} \right) \times \bar{f} \left(\frac{1}{N} \sum_{k=1}^N DR_k, DR_{tar} \right) \quad (5.3.3)$$

其中, f 代表适应度函数, 用于评价算法的性能, 其值越大, 说明系统参数性能越优秀。 p_{\max} 代表最大发射功率, 函数 $\bar{f}(y, \alpha)$ 可以定义为

$$\bar{f}(y, \alpha) = \frac{1}{2} \left(\tanh \left(\sigma \left(\log(y / \alpha) - \eta \right) \right) + 1 \right) \quad (5.3.4)$$

其中, σ 和 η 是两个常数, 将其设置为 $\sigma = 1.65$, $\eta = -0.9$ 。这种参数设置使得当 $y / \alpha = 0$ 时, 函数 $\bar{f}(y, \alpha)$ 趋近于 0; 当 $y / \alpha = 1$ 时, 函数 $\bar{f}(y, \alpha)$ 趋近于 1。函数 $\bar{f}(y, \alpha)$ 的收敛曲线如图 5.3.1 所示。从图可以得知, 函数 $\bar{f}(y, \alpha)$ 取值被限制在 $[0, 1]$ 之间。当 y 值小于 α 时, 函数 $\bar{f}(y, \alpha)$ 取值随着 y 的增加而显著增加; 当 y 值趋近于或超过 α 时, 函数 $\bar{f}(y, \alpha)$ 取值随着 y 的增加而缓慢增加并接近 1。由此可以看出, 函数 $\bar{f}(y, \alpha)$ 在式 (5.3.3) 中起到了惩罚因子的作用, 即当用户的比特错误率和数据速率没有达到目标值时, 对其施以较大的惩罚; 当用户的比特错误率和数据速率达到目标值时, 函数 $\bar{f}(y, \alpha)$ 值接近于 1, 则用户的发射功率发挥主要作用。这样的设计使得适应度函数取值越大, 所获得的通信性能越优秀。由此可以看出, 所设计的参数优化模型能够保证在满足用户服务质量的前提下实现严格的功率控制。

图 5.3.1 函数 $\bar{f}(y, \alpha)$ 的收敛曲线

5.3.2 量子细菌觅食算法的绿色认知无线电参数调整^[29]

在本节中，将所提出的 QBFA 应用于求解绿色认知无线电参数调整问题，并把式（5.3.3）中的函数作为 QBFA 的适应度函数，则绿色认知无线电参数调整的过程等同于对适应度函数进行寻优的过程，也就是寻找使得适应度函数值最大的量子位置和位置的过程。

基于量子细菌觅食算法的参数调整方法可被描述如下。

Step 01 初始化量子细菌种群，包括量子细菌的量子位置和位置。初始化 QBFA 算法参数，将量子细菌的二进制位置映射成认知无线电的系统参数。计算每个量子细菌位置所对应的适应度函数值。初始化每个量子细菌的局部最优位置和整个量子细菌种群的全局最优位置。

Step 02 迁徙操作循环： $l = l + 1, q = 0$ 。

Step 03 复制操作循环： $q = q + 1, j = 0$ 。

Step 04 量子趋向性操作循环： $j = j + 1$ ，更新每个量子细菌的量子旋转角、量子位置和位置，并将二进制位置映射为认知无线电系统参数，计算每个量子细菌位置所对应的适应度函数值。更新每个量子细菌的局部最优位置和整个量子细菌种群的全局最优位置。

Step 05 如果 $j < N_c$ ，则返回 **Step 04** 继续进行趋向性操作。

Step 06 进行复制操作。

Step 07 如果 $q < N_{re}$ ，则进入 **Step 03**，继续完成下一代复制操作，开始进行新一轮的量子趋向性运动。

Step 08 如果 $l < N_{ed}$ ，则进行迁徙操作。

Step 09 如果算法达到了最大迭代次数，则迭代终止，并输出最优结果；否则，返回 **Step 02**。

5.3.3 试验仿真

将量子细菌觅食算法 (MQBCO) 用于解决绿色认知无线电参数调整问题，分别与基于遗传算法 (GA)、量子遗传算法 (QGA) 和粒子群算法 (PSO) 的参数调整方法进行比较。仿真基于一个包含 32 个子载波的多载波系统。为每个子载波分配一个 [0,1] 间的随机数用于模拟信道的衰落。发射功率用 6 位二进制比特进行编码，其范围为 1-65 mW。调制方式用 2 位二进制比特进行编码，采用 4 种调制方式，分别为 4-PSK、16-QAM、32-QAM 和 64-QAM，符号速率为 1 Msps，信道噪声为加性高斯白噪声，噪声基底为 -90 dBm。这 4 种算法的种群规模及最大迭代次数设置均相同，种群规模设置为 20，最大迭代次数设置为 1000。PSO 参数设置：两个学习因子均设为 $2^{[28]}$ 。GA 参数设置：交叉概率为 0.8，变异概率为 0.02。QGA 参数设置：量子旋转角从初始代的 0.1π 线性递减至最后一代的 $0.005\pi^{[6]}$ 。QBFA 参数设置情况如下： $N_c=500$ ， $N_{ed}=2$ ， $N_{re}=1$ ， $e_1=0.06$ ， $e_2=0.015$ ， $p_{ed}=0.1$ ，最大迭代次数 $N_c \cdot N_{re} \cdot N_{ed}=1000$ 。实验结果为 100 次实验仿真的平均值。

图 5.3.2 和图 5.3.3 给出了两种通信模式下的适应度函数收敛曲线。模式一适用于多媒体通信： $BER_{tar}=10^{-6}$ ， $DR_{tar}=4 \times 10^6$ bps；模式二适用于低速率通信： $BER_{tar}=10^{-6}$ ， $DR_{tar}=2.5 \times 10^6$ bps。

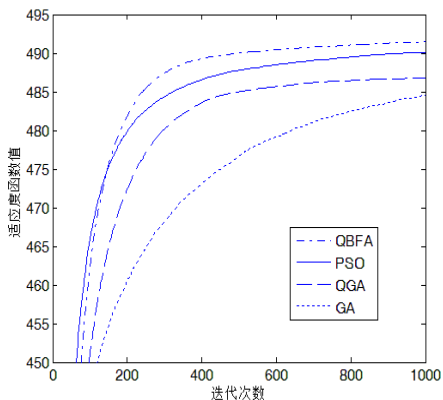


图 5.3.2 多媒体通信模式的收敛曲线

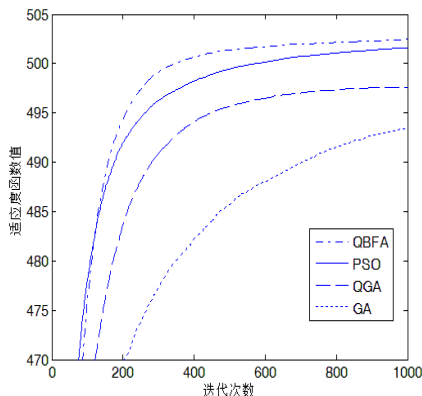


图 5.3.3 低速率通信模式的收敛曲线

从图 5.3.2 和图 5.3.3 中可以看出, QBFA 与 GA、QGA、PSO 相比具有更好的收敛效果, 而其他三种算法均陷入局部收敛。也就是说 QBFA 的收敛性能在 4 种算法中是最优秀的。

图 5.3.4~图 5.3.7 给出了在模式二, 即低速率通信模式下, 4 种算法所获得的参数调整结果。从图中可以看到每个子载波的数据速率、发射功率和比特错误率情况, 在相同的信道衰减下, 所提出的 QBFA 算法能够在用户服务质量得到保证的前提下实现低功耗工作, 而其他三种算法却不能合理地功率控制和用户需求之间进行均衡调节。所提出的基于量子细菌觅食的参数调整方法的高效性主要归功于量子细菌觅食算法内在的搜索机制。

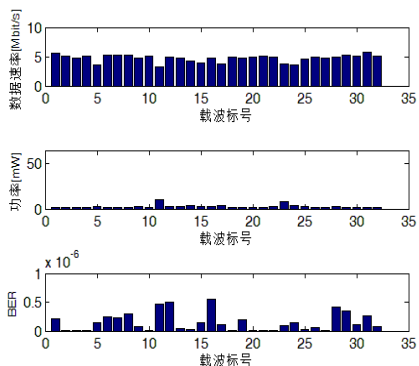


图 5.3.4 遗传算法获得的参数调整结果

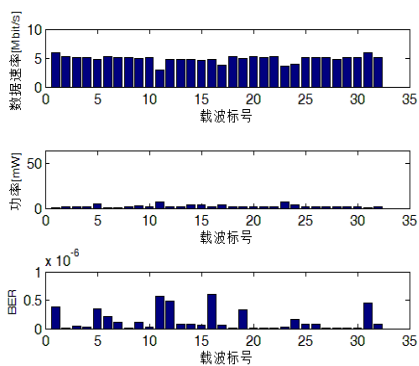


图 5.3.5 量子遗传算法获得的参数调整结果

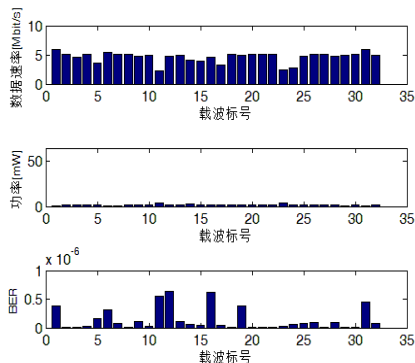
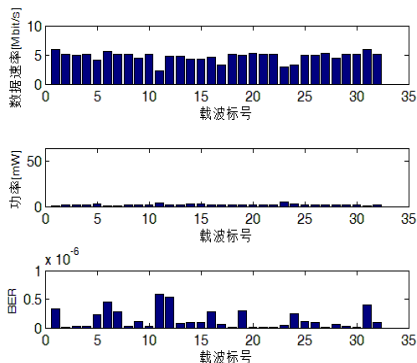


图 5.3.6 粒子群算法获得的参数调整结果 图 5.3.7 量子细菌觅食算法获得的参数调整结果

图 5.3.8 和图 5.3.9 分别仿真了当 $BER_{tar} = 10^{-6}$ 和 $BER_{tar} = 10^{-4}$ 时,随着 DR_{tar} 变化,4 种算法所求得总发射功率的收敛曲线。从图中可以看出,当 BER_{tar} 保持不变时,总发射功率随着 DR_{tar} 增加大致呈现单调递增的趋势。尽管 DR_{tar} 不断改变,但所提出的 QBFA 算法均能获得最小的总发射功率。

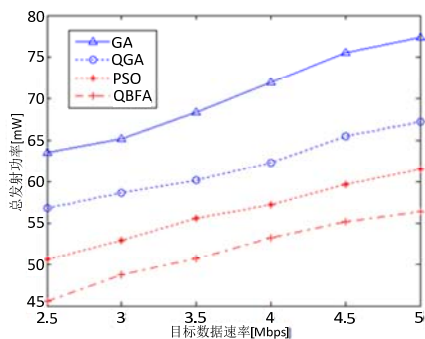


图 5.3.8 当 $BER_{tar} = 10^{-6}$ 时,总发射功率的性能曲线

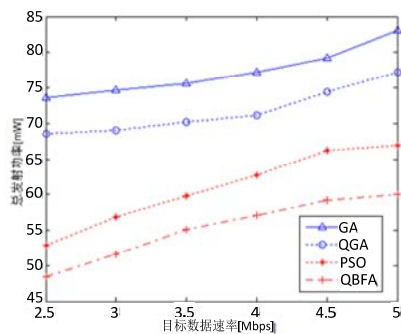


图 5.3.9 当 $BER_{tar} = 10^{-4}$ 时,总发射功率的性能曲线

图 5.3.10 和图 5.3.11 仿真了当 $DR_{tar} = 3\text{Mbps}$ 和 $DR_{tar} = 4\text{Mbps}$ 时,随着 BER_{tar} 变化,4 种算法所求得总发射功率的收敛曲线。从图中可以看出,当 DR_{tar} 保持不变时,总发射功率随着 BER_{tar} 增加大致呈现单调递减的趋势。图 5.3.10 和图 5.3.11 中所获得的最低总发射功率均远小于单个子载波所允许的最大发射功率,而其他三种算法并没有获得足够低的功率。所提出的基于量子细菌觅食

的参数调整方法在不同的用户需求下，能使用最低的发射功率获得相应的系统性能要求，因此所提方法的优越性和稳定性得以验证。

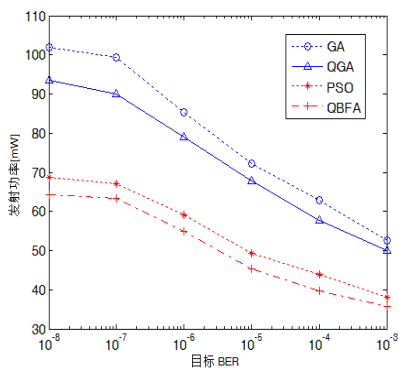


图 5.3.10 当 $DR_{tar}=3\text{Mbps}$ 时，总发射功率的性能曲线

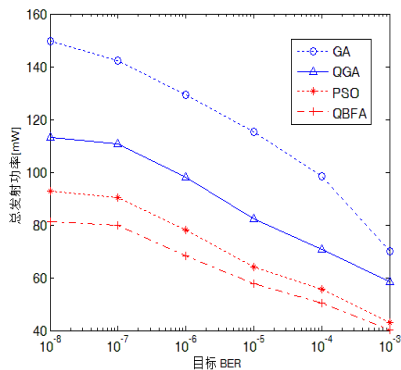


图 5.3.11 当 $DR_{tar}=4\text{Mbps}$ 时，总发射功率的性能曲线

所提算法获得的最小总发射功率可低至 40mW 左右，远小于一个单载波所允许的最大发射功率，所提参数调整方法的节能效果是显而易见的。算法中量子计算理论的量子编码使得量子细菌种群能够保持多样性，且量子旋转门操作可以利用已有量子细菌的经验信息来指引种群向最优位置演进，所以所提出的 QBFA 能获得较好的收敛性能，且保证了基于 QBFA 的参数调整方法是高效而可靠的。

5.4 小结

本章首先介绍了三种基于经典智能计算算法的决策引擎，然后重点给出了基于膜量子蜂群算法的决策引擎方法和基于量子细菌觅食算法的绿色认知无线电的参数调整，可以解决现有认知决策引擎和参数调整技术面临的一些问题。基于膜量子蜂群算法的认知无线电决策引擎方法能够克服经典算法易早熟收敛的缺点，可以获得较高的收敛精度，且实时性较好。最后主要分析和研究了绿色认知无线电的参数调整问题，并将量子细菌觅食算法用于解决绿色认知无线

电决策引擎参数调整问题, 通过与遗传算法、量子遗传算法和粒子群算法进行对比, 在绿色认知无线电系统中证实了量子细菌觅食算法进行参数调整的优越性。

参考文献

- [1] Nguyen V T, Villain F, Le Guillou Y. Cognitive radio systems: Overview and challenges. 3rd IEEE International Conference on Awareness Science and Technology (iCAST), 2011: 497-502 .
- [2] Rieser C J, Rondeau T W, Bostian C W, et al. Cognitive radio testbed: further details and testing of a distributed genetic algorithm based cognitive engine for programmable radios. IEEE Military Communications Conference, 2004: 1437-1443.
- [3] Newman T R, Barker B A, Wyglinski A M, et al. Cognitive engine implementation for wireless multicarrier transceiver. Wireless communications and mobile computing, 2007, 7(9): 1129-1142.
- [4] Newman T R, Evans J B. Parameter sensitivity in cognitive radio adaptation engines. 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2008:756-760.
- [5] Newman T R, Rajbanshi R, et al. Population adaptation for genetic algorithm-based cognitive radios. Mobile Networks and Applications, 2008, 13(5): 442-451.
- [6] 赵知劲, 郑仕链, 尚俊娜, 等. 基于量子遗传算法的认知无线电决策引擎研究.物理学报, 2007, 56 (11): 6760-6766.
- [7] 郑仕链, 赵知劲. 基于模拟退火遗传算法的认知无线电决策引擎研究. 计算机仿真, 2005, 1 (25): 192-196.

- [8] 赵知劲, 徐世宇, 郑仕链, 等. 基于二进制粒子群算法的认知无线电决策引擎. 物理学报, 2009, 58 (7): 5118-5125.
- [9] Zhang J, Zhou Z, Gao W, et al. Cognitive radio adaptation decision engine based on binary quantum-behaved particle swarm optimization. 6th IEEE International ICST Conference on Communications and Networking in China (CHINACOM), 2011: 221-225.
- [10] 李晨琬. 基于量子群智能的认知无线电关键技术研究. 哈尔滨工程大学硕士学位论文, 2015.
- [11] Guo Q W, Jin L L, Min Z, et al. Solving performance optimization problem of cognitive radio with multi objective evolutionary algorithm. Computer Engineering and Applications, 2007, 43(20):159-162.
- [12] Yong L, Hong J, Qing H Y. Design of cognitive radio wireless parameters based on multi-objective immune genetic algorithm. IEEE International Conference on Communications and Mobile Computing, 2009, 1: 92-96.
- [13] Naeem M, Illanko K, Karmokar A, et al. Optimal power allocation for green cognitive radio: fractional programming approach. Communications, IET, 2013, 7(12): 1279-1286.
- [14] Gao Hongyuan, Liu Dandan, and Du Yanan. Parameter optimization based on evolutionary algorithms for green cognitive radio. Smart Computing Review, 2015, 5(5):388-399.
- [15] 赵知劲, 郑仕链, 尚俊娜. 认知无线电技术. 北京: 科学出版社, 2008.
- [16] Buddhikot M M, Kolodzy P, Miller S, et al. DIMSUMnet: new directions in wireless networking using coordinated dynamic spectrum. Sixth IEEE International Symposium on World of Wireless Mobile and Multimedia Networks, 2005: 78-85.
- [17] Chaiyaratana N, Zalzala A M S. Recent developments in evolutionary and

- genetic algorithms: theory and applications. Second International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications, 1997:270-277.
- [18] Zalzal A M, Peter J F. Genetic algorithms in engineering systems. IET, 1997.
- [19] Xu H Y, Zhou Z. Hill-climbing genetic algorithm optimization in cognitive radio decision engine. 15th IEEE International Conference on Communication Technology (ICCT), 2013:17-19.
- [20] Gao Hongyuan, Li Chenwan. Membrane-inspired quantum bee colony optimization and its applications for decision engine. Journal of Central South University, 2014, 21(5):1887-1897.
- [21] Păun G. Computing with membranes. Journal of Computer and System Sciences, 2000, 61(1): 108-143.
- [22] Gao Hongyuan, Cao Jinlong. Membrane-inspired quantum shuffled frog leaping algorithm for spectrum allocation. Journal of Systems Engineering and Electronics, 2012, 23(5): 679-688.
- [23] 高洪元, 曹金龙. 量子蜂群算法及其在认知频谱分配中的应用. 中南大学学报 (自然科学版), 2012, 43 (12): 4744-4749.
- [24] Palicot J, Louët Y, Mroué M. Peak to average power ratio sensor for green cognitive radio. IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2010: 2669-2674.
- [25] Le T M, Lasaulce S, Hayel Y, et al. Green power control in cognitive wireless networks. IEEE Transactions on Vehicular Technology, 2013, 62(4): 1741-1754.
- [26] Ramamonjison R, Haghnegahdar A, Bhargava V. Joint optimization of clustering and cooperative beamforming in green cognitive wireless networks. 2014, 13(2): 982-997.

- [27] Naeem M, Illanko K, Karmokar A, et al. Iterative power allocation for downlink green cognitive radio network. 2012 IEEE Globecom Workshops (GC Wkshps), 2012: 163-167.
- [28] 郑仕链, 杨小牛. 绿色认知无线电自适应参数调整. 物理学报, 2012, 61(14): 148402.
- [29] Gao Hongyuan, Li Chenwan. Quantum-inspired bacterial foraging algorithm for parameter adjustment in green cognitive radio. Journal of Systems Engineering and Electronics, 2015, 26(5):897-907.

基于量子群智能的频谱分配

认知无线电系统的频谱分配必须考虑三方面的问题，即：（1）认知用户对授权用户的干扰问题；（2）认知用户间的干扰问题；（3）整个认知无线网络系统的效益与公平性问题。基于此，许多分配模型被提出，如图论着色模型^[1]、干扰温度模型^[2]、定价拍卖模型^[3]和博弈论模型^[4]。频谱分配技术也有很多类，按频谱接入分类可分为完全受限频谱分配和部分受限频谱分配；按网络结构可分为集中式频谱分配和分布式频谱分配；按合作方式可分为合作式频谱分配和非合作式频谱分配^[5]。

图论着色模型是一种受到广泛关注的较为成熟的频谱分配模型，也可被看作一种组合优化问题，可利用一些智能算法来求解，如遗传算法^[6]、量子遗传算法^[7]和粒子群优化算法^[8]等。然而现有经典智能计算方法的收敛速度和收敛精度存在相互制约的缺点，并不能保证在合理的时间内获得较为理想的解，需要改进才可获得好的性能和速度^[9-12]。此外，上述经典智能算法都是针对单目标优化问题设计的，面对认知无线电频谱分配这个复杂的多目标优化问题就无能为力了，也无法使认知无线电系统的网络效益和公平性达到联合最优的状态，因此借鉴智能计算理论的最新进展来设计新的多目标优化算法有重要的理论价值和实践意义^[13]。

为了获得有效的频谱分配方案,本章首先介绍了一种使用量子粒子群算法进行单目标频谱分配的方法,虽然可以分别求出不同网络效益函数的最优值,但不能同时考虑效益和公平性。然后提出了多目标膜量子蜂群优化算法的多目标频谱分配方法,在求解多目标频谱分配问题的过程中使网络效益和用户间公平性同时达到最佳。

6.1 频谱分配模型

本节介绍的基于图论着色的认知无线电频谱分配模型是合作的分布式完全受限频谱分配,即通过多个认知用户之间相互交换分配信息、协商频谱分配,合力达到全局最优的频谱分配目的。

6.1.1 图论着色模型

用于频谱分配的图论着色模型在移动蜂窝通信中是一种较为成熟的模型,近年来也被学者引入认知无线电技术中继续深入研究^[14]。在认知无线网络中,授权用户使用信道情况是随机性行为,认知用户通过感知环境获得可用的空闲频谱,并在考虑干扰限制等条件下对其进行分配。

在建立图论着色的频谱分配模型中,利用认知用户和授权用户的信息,认知无线网络可被抽象成一张拓扑图,其中每个顶点代表一个认知用户,每个可用频谱用一种颜色表示。如图 6.1.1 所示为一个包含 4 个授权用户和 6 个认知用户的拓扑示意图。图中的 6 个认知用户用 1~6 来表示,4 个授权用户用 I~IV 来表示,工作范围用虚线圆圈表示,这些授权用户的授权频段由信道 A~C 来表示。从图中可见,授权用户 I 和 III 共用信道 A,因为这两个授权相距较远,工作范围没有覆盖,所以不会产生干扰。认知用户采用机会式频谱接入,当授权用户在其授权频段工作时,在其工作范围内的认知用户不可以使用相应频段。由图可知,认知用户 1 位于授权用户 I 的工作范围内,则其不能使用信道 A;

认知用户 5 位于授权用户 II 和 IV 的工作范围内，则只能选择信道 A。以此类推可以得到各个认知用户的可用频谱（在图 6.1.1 中用括号里的内容表示）。除此之外，在通信过程中认知用户之间相互干扰的情况也应该被考虑。当两个相距较近的认知用户在某一时刻使用同一频段时，就会产生彼此干扰的情况。为避免此种干扰现象，在图论着色模型中，两个认知用户之间的连线用于表示这两个用户不能在同一时刻使用相同的信道。

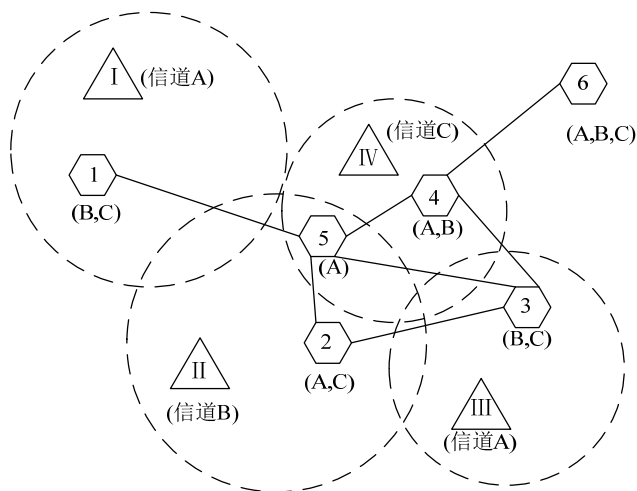


图 6.1.1 网络拓扑示意图

下面将给出认知无线电中用于频谱分配的图论着色数学模型。

考虑有一个包含 N 个认知用户的认知网络，这些认知用户竞争使用 M 个可用的正交频段。根据授权用户与认知用户之间的位置和各自的工作范围（通信范围）等信息可以建立可用频谱矩阵与效益矩阵。

可用频谱矩阵 $\mathbf{L} = \{l_{n,m} \mid l_{n,m} \in \{0,1\}\}_{N \times M}$ 可以表示出每个频段对各认知用户的可用性信息，它是一个 N 行 M 列的矩阵。如果认知用户 n 的工作范围在某一授权用户所覆盖的工作范围外，那么此授权用户的授权频段 m 对于用户 n 是可用的， $l_{n,m}=1$ ；反之，则不可用， $l_{n,m}=0$ 。 $l_{n,m}$ 的值可依据如下数学表达式确定：若 $d_s(n,m) < d_{\min}$ 则 $l_{n,m}=1$ ；否则 $l_{n,m}=0$ ，其中 $d_s(n,m) = \text{Dist}(n,x) - d_p(x,m)$ ， $d_p(x,m)$ 为使用频段 m 的授权用户 x 的工作范围的半径。 $\text{Dist}(n,x)$

表示认知用户 n 和授权用户 x 间的距离。一般情况下, 认知用户的通信覆盖范围的半径 d_s 具有最大值和最小值, 即 $d_s \in [d_{\min}, d_{\max}]$ 。

假设用户的信噪比是 $d_s(n, m)$ 的函数, 用户获得的效益用最大传输带宽来表示: $b_{n,m} = \log(1 + f(d_s(n, m)))$, $d_{\min} \leq d_s(n, m) \leq d_{\max}$ 。如果 $l_{n,m} = 0$, 那么 $b_{n,m} = 0$ 。

干扰矩阵用来表示不同的认知用户在使用相同频段时产生干扰的情况, 可以用矩阵 $\mathbf{C} = \{c_{n,k,m} \mid c_{n,k,m} \in \{0, 1\}\}_{N \times N \times M}$ 来表示。若 $c_{n,k,m} = 1$, 则认知用户 n 和认知用户 k 在某一时刻共同使用频段 m 时会彼此干扰。干扰矩阵会受限于可用频谱矩阵, 具体为 $c_{n,k,m} \leq l_{n,m} \times l_{k,m}$ 。在 $n = k$ 的情况下, $c_{n,k,m} = 1 - l_{n,m}$, 说明此时干扰矩阵只能通过可用频谱矩阵 \mathbf{L} 确定。

无干扰分配矩阵 $\mathbf{A} = \{a_{n,m} \mid a_{n,m} \in \{0, 1\}\}_{N \times M}$ 可给出具体的频谱分配方法。若频道 m 被分配给认知用户 n , 则 $a_{n,m} = 1$ 。 $\mathbf{A} = \{a_{n,m} \mid a_{n,m} \in \{0, 1\}\}_{N \times M}$ 必须符合分配过程中所涉及的干扰约束条件: $a_{n,m} + a_{k,m} \leq 1$, 若 $c_{n,k,m} = 1, \forall 1 \leq n, k \leq N, 1 \leq m \leq M$ 。

频谱分配的数学模型可以被看作为对图 $\mathbf{G} = (\mathbf{U}, \mathbf{E}_c, \mathbf{L}_B)$ 进行着色的问题。 \mathbf{U} 表示图 \mathbf{G} 的各顶点, 代表待分配频谱的各认知用户; \mathbf{L}_B 表示所有顶点可供选择的颜色集合和相应权重; \mathbf{E}_c 表示图的边集情况, 并由干扰约束条件集合 \mathbf{C} 确定。当且仅当 $c_{n,k,m} = 1$ 时, 有一条颜色为 m 的边存在于两个不同的顶点 (用户) $u, v \in \mathbf{U}$ 之间。对于符合无干扰分配条件的频谱分配方案, 可被理解为: 当有一条颜色为 m 的边在两个顶点之间时, 不可以把这两个顶点同时着 m 色。

6.1.2 单目标频谱分配

基于 6.1.1 节所介绍的图论着色频谱分配模型, 设 $r_n = \sum_{m=1}^M a_{n,m} \cdot b_{n,m}$ 为认知用户 n 使用为其分配的频段所得到的效益, 效益矩阵 $\mathbf{R} = \left\{ r_n = \sum_{m=1}^M a_{n,m} \cdot b_{n,m} \right\}_{N \times 1}$ 包含所有认知用户所获得的效益。

频谱分配问题可以被看作离散优化问题，即从所有可行的频谱分配方法中选择一种分配方案使得通信系统可以达到某种预期的目标，此目标可以由不同的网络效益来代表。可用 $U(\mathbf{A})$ 表示网络效益函数，设 \mathbf{A}^* 为满足某目标的最优频谱分配方案，则 \mathbf{A}^* 满足： $\mathbf{A}^* = \arg \max_{\mathbf{A} \in \Lambda(L, C)_{N \times M}} U(\mathbf{A})$ 。常用的三种网络效益函数如下。

(1) 基于最大和网络效益 (Max-Sum-Reward, MSR), $U(\mathbf{A})$ 定义为

$$U_{\text{MSR}}(\mathbf{A}) = \sum_{n=1}^N r_n = \sum_{n=1}^N \sum_{m=1}^M a_{n,m} \cdot b_{n,m} \quad (6.1.1)$$

此网络效益的目标是最大化 CRS 的总网络效益，但是该定义并没有充分考虑用户之间的公平性。

(2) 基于最大化最小网络效益 (Max-Min-Reward, MMR), $U(\mathbf{A})$ 定义为

$$U_{\text{MMR}}(\mathbf{A}) = \min_n r_n = \min_n \sum_{m=1}^M a_{n,m} \cdot b_{n,m} \quad (6.1.2)$$

其目标是使得获得最小网络效益的用户的网络效益最大化，这种网络效益函数保证了受限用户的公平性。

(3) 基于最大比例公平网络效益 (Max-Proportional-Fair, MPF), $U(\mathbf{A})$ 定义为

$$U_{\text{MPF}}(\mathbf{A}) = \sum_{n=1}^N \log(r_n) = \sum_{n=1}^N \log \left(\sum_{m=1}^M a_{n,m} \cdot b_{n,m} \right) \quad (6.1.3)$$

此网络效益函数同样体现了频谱分配过程中用户之间的公平性，且保证了系统具有一定的吞吐量。其实质是如果存在另外一种频谱分配方案 \mathbf{A}' ，那么其相关的用户效益 r'_n 满足下式： $\sum_{n=1}^N \frac{r'_n - r_n}{r_n} = 0$ 。

6.1.3 多目标频谱分配

6.1.2 节介绍了三种常用的效益函数，分别是最大和网络效益 $U_{\text{MSR}}(\mathbf{A})$ 、最大比例公平网络效益 $U_{\text{MPF}}(\mathbf{A})$ 及最大化最小网络效益 $U_{\text{MMR}}(\mathbf{A})$ 。其中 $U_{\text{MPF}}(\mathbf{A})$

和 $U_{\text{MMR}}(\mathbf{A})$ 都可以表示用户间的公平性，但侧重点不同。 $U_{\text{MPF}}(\mathbf{A})$ 保证了每个用户都有一定的效益，在此基础上按照公平的规则进行分配频谱。 $U_{\text{MMR}}(\mathbf{A})$ 保护了贫困用户（获得效益最小的用户），避免其没有信道可用，目的是最大化贫困用户所获得的效益。

为使得最大和网络效益和用户间公平性均达到最优，可建立如下多目标频谱分配数学模型：

$$\max \{U(\mathbf{A}) = [U_{\text{MSR}}(\mathbf{A}) \quad U_2(\mathbf{A})]\} \quad (6.1.4)$$

其中， $U_2(\mathbf{A})$ 代表用户间公平性的网络效益函数， $U_2(\mathbf{A}) \in \{U_{\text{MPF}}(\mathbf{A}), U_{\text{MMR}}(\mathbf{A})\}$ 可根据实际要求选取。 $U_{\text{MSR}}(\mathbf{A}) = \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^M a_{n,m} \cdot b_{n,m}$ ， $U_{\text{MPF}}(\mathbf{A}) = \left(\prod_{n=1}^N \left(\sum_{m=1}^M a_{n,m} \cdot b_{n,m} + 10^{-6} \right) \right)^{\frac{1}{N}}$ ， $U_{\text{MMR}}(\mathbf{A}) = \min_{1 \leq n \leq N} \sum_{m=1}^M a_{n,m} \cdot b_{n,m}$ 。

6.2 基于量子粒子群算法的单目标频谱分配

由于认知无线电频谱分配问题是一个离散优化问题，因而可将单链编码的量子粒子群算法用于设计认知无线电频谱分配方案。由于量子粒子群算法对相应的量子旋转门公式进行简化，降低运算量，因而设计了全新的量子粒子群算法，基于量子粒子群算法的频谱分配方法具有更高的收敛精度和更快的收敛速度。

6.2.1 基于单链量子粒子群算法的认知无线电频谱分配^[15]

所有单链量子粒子群算法的量子速度均被初始化为 $1/\sqrt{2}$ ，并且量子粒子的适应度值即为网络效益。频谱分配过程如下：

Step 01 给出可用频谱矩阵 $\mathbf{L} = \{l_{n,m} \mid l_{n,m} \in \{0,1\}\}_{N \times M}$ ，效益矩阵 $\mathbf{B} = \{b_{n,m}\}_{N \times M}$ ，

干扰矩阵 $C = \{c_{n,k,m} \mid c_{n,k,m} \in \{0,1\}\}_{N \times N \times M}$ ，确定该优化问题的维数为 $l = \sum_{n=1}^N \sum_{m=1}^M l_{n,m}$ ，记录 L 中为 1 的元素， $L_1 = \{(n,m) \mid l_{n,m} = 1\}$ ，其中的元素按 n 递增 m 递增的方式排列。 L_1 中元素的个数即为 l 的值。

Step 02 初始化量子粒子群，包括量子粒子的位置、量子粒子的量子速度和量子粒子的局部最优位置。

Step 03 计算量子粒子的适应度值。首先将量子粒子的位置的第 j 位映射到 $a_{n,m}$ 中，其中 (n,m) 是 L_1 中第 j 个元素，且 $1 \leq j \leq \sum_{n=1}^N \sum_{m=1}^M l_{n,m}$ 。然后对于所有的 m ，寻找所有的 (n,k) 满足 $c_{n,k,m} = 1$ 且 $n \neq k$ 。接着检查 A 矩阵的第 n 行第 m 列的元素和第 k 行第 m 列的元素是否都为 1，如果是，则随机地将其中一个变为 0。可以根据无干扰分配矩阵 A 和效益矩阵 B 计算网络效益。把适应度最优的位置记作全局最优位置。

Step 04 更新量子粒子的量子速度和位置。

Step 05 对于量子粒子的新位置，按照 **Step 03** 调整量子粒子的位置并计算适应度值。

Step 06 更新每个量子粒子的局部最优位置，同时更新全局最优位置。

Step 07 如果进化并没有终止（通常由预先设定的最大迭代次数决定），则返回 **Step 04**；否则，算法终止。

6.2.2 仿真结果分析

1. 量子粒子群算法用于认知无线电频谱分配

修改最大和网络效益目标函数为

$$U_{\text{MSR}}(A) = \frac{1}{N} \sum_{n=1}^N \sum_{m=1}^M a_{n,m} \cdot b_{n,m} \quad (6.2.1)$$

这就是说，在以后的仿真中使用平均效益而不是网络总效益。

如果某个认知用户的效益为 0，即存在贫困用户，则式 (6.1.3) 的目标函数为 0。因而修改最大比例公平网络效益目标函数为

$$U_{\text{MPF}}(\mathcal{A}) = \left(\prod_{n=1}^N \left(\sum_{m=1}^M a_{n,m} \cdot b_{n,m} + 1 \times 10^{-6} \right) \right)^{\frac{1}{N}} \quad (6.2.2)$$

意味着每个用户都有初始效益，即 1×10^{-6} 。

敏感图论着色 (CSGC) 的标号方法为非协作式 (NSUM)，遗传算法的交叉概率为 0.8，变异概率为 0.01，选择方式为轮盘赌选择，在精英保留时用迄今为止找到的三个最优个体替换本代最差的三个个体^[6]，量子遗传算法量子旋转角在 $0.005\pi \sim 0.1\pi$ 间动态变化，随进化代数线性递减^[6,7]，粒子群算法的个体学习因子和社会学习因子均为 2，即 $c_1 = c_2 = 2$ ，最大速度限制为 4，即 $v_{\max} = 4$ ^[8]，量子粒子群算法的参数设置为 $e_1 = 0.06$ ， $e_2 = 0.03$ ， $c_1 = 0$ 。这是因为在 6.2.1 节 **Step 03** 中存在变异，故在此处将变异概率设置为 0。将 4 种算法，即遗传算法、量子遗传算法、粒子群算法和量子粒子群算法的种群规模设为 20，最大迭代次数设为 1000。

仿真过程中，设置频谱池中可用频道的数目为 30，授权用户的数目为 20，认知用户的数目为 10。从图 6.2.1 中可以看出，与经典算法（遗传算法、量子遗传算法）相比较，量子粒子群算法具有更快的收敛速度和更高的收敛精度，并且收敛精度相对于 CSGC 更高。

从图 6.2.2 中可以看出，尽管遗传算法和量子遗传算法具有较快的收敛速度，但是均陷入了局部收敛；而量子粒子群算法具有较高的收敛精度，能够更好地找到最优解。

令频谱池中频谱的数目逐渐增多，但认知用户和授权用户的数目保持不变（认知用户数目为 10，授权用户数目为 20），研究网络效益的变化情况。仿真结果如表 6.2.1 和表 6.2.2 所示。随着频道用户数目的增多，认知用户的可用频道数目增多，能够从中获得更多的网络效益。因而，随着频道数目的增多，网络效益增大。同时可以看出，对于频谱分配问题，量子粒子群算法在频道数目增多的情况下依旧优于其他 3 种算法。

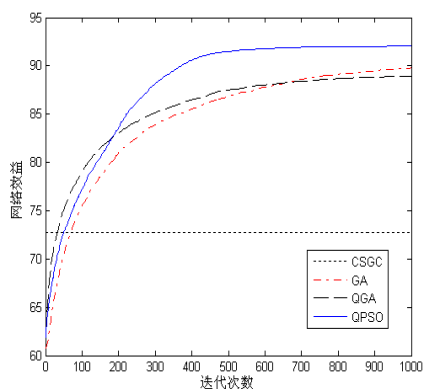


图 6.2.1 基于最大和网络效益的仿真结果

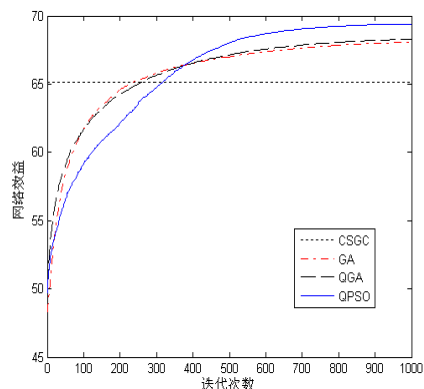


图 6.2.2 基于最大比例公平网络效益的仿真结果

表 6.2.1 随着频道数目增多最大和网络效益的变化情况

算法	频道数目					
	5	10	15	20	25	30
CSGC	7.1335	20.28	34.131	48.13	59.529	72.706
GA	7.4059	23.239	40.251	58.612	72.87	89.79
QGA	7.4097	23.205	40.152	57.915	72.551	88.958
PSO	7.4097	23.263	40.365	58.737	73.683	91.512
QPSO	7.4097	23.275	40.41	59	73.84	92.036

表 6.2.2 随着频道数目增多最大比例公平网络效益的变化情况

算法	频道数目					
	5	10	15	20	25	30
CSGC	0.10473	8.8559	27.224	39.6225	53.999	65.109
GA	0.58484	15.749	29.522	42.349	57.078	68.113
QGA	0.61269	16.012	29.591	42.594	57.229	68.296
PSO	0.65894	16.031	29.716	42.85	57.796	69.166
QPSO	0.66443	16.387	29.96	43.112	57.929	69.397

设置认知用户数增多，而频谱池中的频谱数目和授权用户数目保持不变（频谱数目为 10，授权用户数目为 20），研究网络效益的变化情况。随着认知用户数

目的增多, 认知用户之间的竞争更为激烈, 邻居用户之间产生的干扰也更多, 因而, 平均网络效益随认知用户的增多而减小。同时, 从表 6.2.3 和表 6.2.4 中也可以看出, 量子粒子群算法在认知用户变化时也优于其他解决频谱分配的算法。

表 6.2.3 随着认知用户数目增多最大和网络效益的变化情况

算法	认知用户数					
	5	10	15	20	25	30
CSGC	33.617	20.28	15.914	12.192	10.606	9.1354
GA	35.559	23.239	18.485	14.267	12.621	10.923
QGA	35.554	23.205	18.364	14.139	12.555	10.782
PSO	35.561	23.263	18.506	14.312	12.726	11.027
QPSO	35.561	23.275	18.528	14.359	12.74	11.051

对比表 6.2.3 和表 6.2.4 也可以发现, 在认知用户数目、授权用户数目和可用频道数目相同时, 认知用户得到的最大和网络效益大于最大比例公平网络效益。

表 6.2.4 随着认知用户数目增多最大比例公平网络效益的变化情况

算法	认知用户数					
	5	10	15	20	25	30
CSGC	24.366	8.8559	2.1298	0.54978	0.17299	0.071994
GA	27.216	15.749	9.1972	5.6505	2.4122	0.98789
QGA	27.293	16.012	9.3451	5.5749	2.1378	0.89429
PSO	27.349	16.031	9.5655	5.7752	2.7884	1.1591
QPSO	27.451	16.387	9.754	6.3373	3.1676	1.5145

6.3 基于多目标膜量子蜂群的多目标频谱分配^[16]

为了解决单目标频谱分配方法所存在的问题, 本节提出了多目标膜量子蜂群优化算法, 演化过程和方程以最大值多目标优化为例给出, 在求解多目标频谱分配问题的过程中使最大和网络效益和用户间公平性同时达到最佳数值。所

提算法基于非支配解排序^[17,18]，采用量子蜂群的演化方法，进而于表层膜内求得均匀分布的 Pareto 前端解。该算法在膜结构框架下使用量子蜂群作为演化群体^[19]，扩展了量子蜂群的应用范围，不仅可求解出单目标认知无线电频谱分配方案，而且可得到多目标频谱分配的 Pareto 前端解。

6.3.1 膜量子蜂群的基本演进规则

可以用一系列量子位代表蜜蜂的量子位置，第 i 只蜜蜂的量子位置可表示为

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} & v_{i2} & \cdots & v_{iD} \\ \beta_{i1} & \beta_{i2} & \cdots & \beta_{iD} \end{bmatrix} \quad (6.3.1)$$

其中， $|v_{id}|^2 + |\beta_{id}|^2 = 1$ ， $d = 1, 2, \dots, D$ 。为便于计算，将 v_{id} 和 β_{id} 定义为 $0 \leq v_{id} \leq 1$ ， $0 \leq \beta_{id} = \sqrt{1 - (v_{id})^2} \leq 1$ ，量子位置可简化为 $\mathbf{v}_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$ 。

膜量子蜂群优化通过对蜂群中每只蜜蜂的量子位置进行更新完成算法的演进过程。 θ_{id}^{t+1} 代表第 $t+1$ 次迭代的量子旋转角， v_{id}^t 可以使用简化的量子旋转门进行更新。第 i 只蜜蜂的第 d 个量子位 v_{id}^t 可使用如下公式进行更新

$$v_{id}^{t+1} = |v_{id}^t \cos(\theta_{id}^{t+1}) - \sqrt{1 - (v_{id}^t)^2} \sin(\theta_{id}^{t+1})| \quad (6.3.2)$$

量子非门可使用下述公式进行模拟

$$v_{id}^{t+1} = \sqrt{1 - (v_{id}^t)^2} \quad (6.3.3)$$

在膜量子蜂群优化 (MQBCO) 中，共有 h 只蜜蜂被均匀分配到基础膜和表层膜中。每个膜内的蜜蜂群体由工蜂和观察蜂构成，数量各占一半。每只工蜂或者观察蜂都对应一个量子位置，对量子位置进行测量可以得到相应的食物源位置。每只蜜蜂根据以往的觅食经验和整个蜂群的信息交流，在所有可能的解空间内搜索最优秀的食物源位置。

在 MQBCO 中，每只蜜蜂的量子位置经过测量可得到食物源位置，可用 D 维取值属于集合 $\{0,1\}$ 的二进制比特代表。第 i 只蜜蜂的食物源位置可以表示为 $\mathbf{x}_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t]$ ，该位置代表了待优化问题的一个潜在的解。第 i 只蜜蜂在

第 t 次迭代过程中搜索到的蜜源含量最高的位置，记作局部最优位置 $p_i^t = [p_{i1}^t, p_{i2}^t, \dots, p_{id}^t] (i=1, 2, \dots, h)$ 。

在 MQBCO 中, $R_r (0 \leq r \leq f)$ 为第 r 个膜的演进规则。将所有蜜蜂平均分配到 f 个基础膜和 1 个表层膜中，则每个膜内都有 $0.5h/(f+1)$ 只工蜂和 $0.5h/(f+1)$ 只观察蜂。将蜜蜂用 $1, 2, \dots, h$ 进行标号，则第 $r(1 \leq r \leq f)$ 个基础膜内的蜜蜂标号为 $i \in \left[\frac{hr}{f+1} + 1, \frac{h(r+1)}{f+1} \right] (i \in \mathbb{Z})$ ，每个基础膜内前一半标号代表工蜂，后一半标号代表观察蜂，表层膜内的蜜蜂标号为 $i \in \left[1, \frac{h}{f+1} \right] (i \in \mathbb{Z})$ ，其中 \mathbb{Z} 表示整数集。

在第 $r(1 \leq r \leq f)$ 个基础膜内，第 i 只工蜂的第 d 维量子位可以使用如下方程更新

$$\theta_{id}^{t+1} = e_1(p_{id}^t - x_{id}^t) + e_2(g_{rd}^t - x_{id}^t) \quad (6.3.4)$$

$$v_{id}^{t+1} = \begin{cases} \sqrt{1 - (v_{id}^t)^2}, & \theta_{id}^{t+1} = 0 \text{ 且 } \gamma_{id}^{t+1} < c_1 \\ |v_{id}^t \cos(\theta_{id}^{t+1}) - \sqrt{1 - (v_{id}^t)^2} \sin(\theta_{id}^{t+1})|, & \text{其他} \end{cases} \quad (6.3.5)$$

其中， $d=1, 2, \dots, D$ ； e_1 和 e_2 是两个常数，分别代表第 i 只工蜂的局部最优位置 and 所在基础膜内全局最优位置对量子旋转角确定的权重数值； $g_r^t = [g_{r1}^t, g_{r2}^t, \dots, g_{rd}^t]$ 是第 r 个膜至第 t 代所发现的全局最优位置； $\gamma_{id}^{t+1} \in [0, 1]$ 是一个均匀分布的随机数； $c_1 \in [0, 1/D]$ 为量子旋转角等于 0 时的量子位的翻转概率。

在第 $r(1 \leq r \leq f)$ 个基础膜内每只观察蜂的量子旋转角及量子位置可分别使用式 (6.3.6) 和 (6.3.7) 进行更新。

$$\theta_{id}^{t+1} = e_3(p_{id}^t - x_{id}^t) + e_4(p_{jd}^t - x_{id}^t) + e_5(g_{rd}^t - x_{id}^t) \quad (6.3.6)$$

$$v_{id}^{t+1} = \begin{cases} \sqrt{1 - (v_{id}^t)^2}, & \theta_{id}^{t+1} = 0 \text{ 且 } \gamma_{id}^{t+1} < c_2 \\ |v_{id}^t \cos(\theta_{id}^{t+1}) - \sqrt{1 - (v_{id}^t)^2} \sin(\theta_{id}^{t+1})|, & \text{其他} \end{cases} \quad (6.3.7)$$

其中， $d=1, 2, \dots, D$ ；常数 e_3 、 e_4 和 e_5 分别代表量子旋转角受观察蜂的局部

极值、所选择工蜂的局部极值和该膜中的全局极值影响的权重数值； $c_2 \in [0, 1/D]$ 代表量子旋转角等于 0 时量子位的翻转概率； $\mathbf{p}_j^t = [p_{j1}^t, p_{j2}^t, \dots, p_{jD}^t]$ 代表第 r 个膜内第 j 只工蜂的局部最优位置。第 $i \left(\frac{h(r+0.5)}{f+1} + 1 \leq i \leq \frac{h(r+1)}{f+1} \right)$ 只观察蜂根据轮盘赌规则依概率选择工蜂 j 。 $F_r(\mathbf{p}_j^t)$ 为第 j 只工蜂在第 r 个基础膜内搜索的局部最优位置的适应度函数值，适应度值越大说明该工蜂越优秀，被观察蜂挑选的概率也越大。综上所述，在第 r 个基础膜内的第 $j \left(\frac{hr}{f+1} + 1 \leq j \leq \frac{h(r+0.5)}{f+1} \right)$ 只工蜂被选择的概率为

$$q_j^{t+1} = \frac{F_r(\mathbf{p}_j^t)}{\sum_{l=hr/(f+1)+1}^{(0.5+r)h/(f+1)} F_r(\mathbf{p}_l^t)} \quad (6.3.8)$$

6.3.2 膜量子蜂群的膜框架

基于量子蜂群的膜框架^[20]构成如下。

(1) 有 f 个区域，都存在于标号为 0 的表层膜内，其膜结构可表示为 $[0][1]_1[2]_2 \cdots [f]_f]_0$ 。

(2) 字母表是由量子位置和二进制位置向量构成的集合。

(3) 输出端字母表 \mathbf{T} 由二进制位置向量表示。

(4) 多重集 $\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_f$ 由量子位置集合所构成。每个膜内的蜜蜂数为 $n_i = h/(f+1)$ 。多重集是量子位置的集合，可记作 $\mathbf{w}_0 = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{n_0}\}$ ， $\mathbf{w}_1 = \{\mathbf{v}_{n_0+1}, \mathbf{v}_{n_0+2}, \dots, \mathbf{v}_{n_0+n_1}\}$ ， \dots ， $\mathbf{w}_f = \{\mathbf{v}_{n_0+n_1+\dots+n_{f-1}+1}, \mathbf{v}_{n_0+n_1+\dots+n_{f-1}+2}, \dots, \mathbf{v}_{n_0+n_1+\dots+n_{f-1}+n_f}\}$ 。其中， $\mathbf{v}_i (1 \leq i \leq h)$ 代表第 i 只蜜蜂的量子位置； $\sum_{j=0}^f n_j = h$ ， h 是量子蜂群中的蜜蜂总数。

(5) 规则主要包括演化规则、测量规则和通信规则。

① 演化规则：每个基础膜都以一个单目标函数对膜内蜜蜂的食物源位置进行优劣评判，利用当前膜内蜜蜂的位置信息构建量子旋转角，并模拟量子旋转门对量子位置进行更新[具体演进参考式(6.3.4)~式(6.3.7)];基础膜 r ($1 \leq r \leq f$) 中第 i ($1 \leq i \leq \frac{hr}{f+1} + 1$) 只蜜蜂的局部极值 p_i^t 所使用的更新方法如下

$$p_i^{t+1} = \begin{cases} x_i^{t+1}, & U_r(p_i^t) < U_r(x_i^{t+1}) \\ p_i^t, & U_r(p_i^t) \leq U_r(x_i^{t+1}) \end{cases} \quad (6.3.9)$$

其中, $U_r()$ 为第 r 个基础膜内的蜜源含量函数,也可表示第 r 个目标函数或第 r 个适应度函数,则第 r 个基础膜的全局最优位置更新方法为

$$g_r^{t+1} = \arg \max_{p_i^{t+1}} \{U_r(p_i^{t+1})\}, \frac{hr}{f+1} + 1 \leq i \leq \frac{h(r+1)}{f+1} \quad (6.3.10)$$

表层膜内含有 1 个量子蜂群,其中蜜蜂个体标号为 $1 \leq i \leq \frac{h}{f+1}$ ($i \in \mathbb{Z}$);也包含两个记忆集合,即单目标最优位置集合和非支配精英位置集合。单目标最优位置集合存储着到目前为止搜索到的每个待优化目标的最优位置。第 r ($r=1,2,\dots,f$) 个目标的全局最优位置记作 b_r^{t+1} ,其更新公式如下:

$$b_r^{t+1} = \arg \max_{x_i^{t+1}, b_r^t} \{U_r(x_i^{t+1}), U_r(b_r^t)\}, 1 \leq i \leq \frac{h}{f+1} \quad (6.3.11)$$

非支配精英位置集合存储着多目标算法所获得的较优秀的非支配位置集合,其更新过程为:首先对表层膜内蜜蜂的量子位置进行更新,将对量子位置测量得到的位置和精英记忆位置集合中的位置进行混合,并根据各位置所对应的目标函数值进行非支配位置排序;然后进行位置拥挤度的计算;最后选择非支配位置等级较高(支配等级值越小的支配等级越高)且拥挤度较大的 h_e 个优秀位置作为非支配精英位置集合。

表层膜的量子位置可通过式(6.3.4)、(6.3.5)、(6.3.6)和(6.3.7)来更新,但公式中的位置 p_i^t, p_j^t, g_i^t ($r=0$)都要从非支配精英位置集合前 50%的精英位置中随机选取。

对于多个目标的最大值优化问题,设有位置 $x_i, x_j \in S$, S 为可行位置集合。

如果对于所有的 r , 式 $U_r(\mathbf{x}_i) \leq U_r(\mathbf{x}_j), r=1,2,\dots,f$ 都成立, 即对所有的目标函数都成立, 并且至少有一个严格不等式成立, 则称位置 \mathbf{x}_j 支配 \mathbf{x}_i , \mathbf{x}_j 被称为非支配位置。与之相反, 如果对于所有的 r , $U_r(\mathbf{x}_i) < U_r(\mathbf{x}_j), r=1,2,\dots,f$ 都成立, 且至少有一个严格不等式成立, 则称位置 \mathbf{x}_i 支配 \mathbf{x}_j , \mathbf{x}_i 为非支配位置。如果不满足上述两种情况, 则位置 \mathbf{x}_i 和 \mathbf{x}_j 之间没有任何支配关系^[21]。

非支配位置排序的具体流程可被描述为: 对于每个位置 p , 计算支配位置 p 的位置数目并确定位置 p 所支配的位置集合 S_p 。如果支配位置 p 的数目为 0, 则位置 p 的非支配排序等级为 1。对于每个非支配解排序等级为 1 的位置 p , 记录其支配的位置集合 S_p 中的每个位置 q , 并计算支配 q 的位置数目, 若为 0, 则将位置 q 存于集合 Q 中, 则该位置的非支配等级排序为 2。对 Q 中的每个位置重复上面的过程, 进而可以得到非支配等级排序为 3 的位置集合。重复上面的过程直至获得所有位置的非支配等级。

设属于非支配等级 I 的位置共有 o 个, 可将目标函数值 $U_r(r=1,2,\dots,f)$ 按照升序进行排列, 将目标函数的最小值 $U_r^{\min}(1)$ 和最大值 $U_r^{\max}(o)$ 的位置的拥挤度值定义为 ∞ 。除此之外, 对于每个位置 z , 拥挤度的计算公式为: $I_r(z)_{\text{distance}} = \frac{U_r(z+1) - U_r(z-1)}{U_r^{\max}(o) - U_r^{\min}(1)}$, $z=2,3,\dots,o-1$, $U_r(z+1)$ 和 $U_r(z-1)$ 分别为当前位置的下一个位置的目标函数值和前一个位置的目标函数值。根据上述方法对每个位置的各个目标函数都进行拥挤度计算, 则各个拥挤度分量的和就是该位置的最终拥挤度值。

所设计的多目标算法的目标是向着非支配等级排序为 1 且拥挤度值较大的位置演进, 进而获得均匀的非支配解集。

② 测量规则: 通过对量子位置进行测量可获得代表潜在解的每个蜜蜂的食物源位置。

对第 i 只蜜蜂的量子位置的各量子位进行测量并获得食物源位置的方法如下:

$$x_{id}^{t+1} = \begin{cases} 1, & \lambda_{id}^{t+1} > (v_{id}^{t+1})^2 \\ 0, & \lambda_{id}^{t+1} \leq (v_{id}^{t+1})^2 \end{cases} \quad (6.3.12)$$

其中, $d=1,2,\dots,D$; λ_{id}^{t+1} 代表一个在 $[0,1]$ 之间满足均匀分布的随机数; $(v_{id}^{t+1})^2$ 为量子位 v_{id}^{t+1} 出现“0”状态的概率。

③ 通信规则: 基础膜和表层膜进行信息传递, 第 r ($r=1,2,\dots,f$) 个基础膜将第 r 个目标的全局最优位置 g_r^{t+1} 及其对应的蜜源含量值传送给表层膜, 并与表层膜中单目标最优位置集合中的 b_r^{t+1} 相比较, 如果 g_r^{t+1} 优于 b_r^{t+1} , 则 $b_r^{t+1} = g_r^{t+1}$; 否则 b_r^{t+1} 保持不变。表层膜内的单目标最优位置集合对该目标进行更新后, 将第 r ($r=1,2,\dots,f$) 个目标全局最优 b_r^{t+1} 及其蜜源含量值传送到相应的基础膜中去更新 g_r^{t+1} 。

6.3.3 基于多目标膜量子蜂群算法的多目标频谱分配技术

对 MQBCO 进行初始化时, 所有量子位置的量子比特位都可以被设置为 $1/\sqrt{2}$ 。蜜源含量函数可由网络效益函数代替, 则基于 MQBCO 的多目标频谱分配流程可被描述如下。

Step 01 根据已知条件可求得可用频谱矩阵 $L = \{l_{n,m} | l_{n,m} \in \{0,1\}\}_{N \times M}$ 、效益矩阵 B 和干扰矩阵 C 。将 L 中为 1 的元素记录在矩阵 $L_1 = \{(n,m) | l_{n,m} = 1\}$ 中, L_1 内的元素按 n 递增 m 递增的规则进行排列。计算优化问题的维数 $D = \sum_{n=1}^N \sum_{m=1}^M l_{n,m}$,

即 D 等于 L_1 中元素的个数。

Step 02 对量子蜂群的量子位置和位置进行初始化设置, 将 h 只蜜蜂的量子位置强制设置为 $1/\sqrt{2}$, 并对其进行测量得到其对应的初始食物源位置。量子蜂群中的所有蜜蜂被平均分配到 $1+2$ 个膜中, 该膜的结构可表示为 $[o_1]_1[o_2]_2$, 在第 1 个基础膜内演化第 1 个目标函数, 在第 2 个基础膜内演化第 2 个目标函数, 两个基础膜内演进蜜蜂位置的目标函数不同, 并以这两个目标函数组成的优化向量作为表层膜内算法演进的目标。

Step 03 若新生成的食物源位置不满足无干扰约束条件, 则对位置进行调整使其满足无干扰约束。首先要将蜜蜂食物源位置的第 j 维位置映射到 $a_{n,m}$ 中, 其中 (n,m) 代表 L_1 中的第 j 个元素, $1 \leq j \leq \sum_{n=1}^N \sum_{m=1}^M l_{n,m}$ 。遍历所有的 m , 寻找所有满足式 $c_{n,k,m}=1$ 且 $n \neq k$ 的 (n,k) 。若矩阵 A 的第 n 行第 m 列和第 k 行第 m 列的元素都为 1, 则使用随机的方式将其中一个置 0, 食物源位置的相应位也需要置 0, 则调整后的食物源位置满足约束条件, 可以作为频谱分配的一个潜在解。

Step 04 对于调整后的食物源位置, 根据蜜源函数计算其蜜源量大小。在第 r ($r=1,2$) 个基础膜内利用该膜内的单目标函数计算每只蜜蜂食物源位置对应的蜜源含量, 并确定此基础膜内每只蜜蜂的局部最优位置和该基础膜内的全局最优位置。在表层膜内对每只蜜蜂的食物源位置按照两个目标函数分别计算其蜜源含量, 并将各个目标的蜜源含量最高的位置存入表层膜的单目标最优位置集合中。在表层膜内, 将根据所有位置计算得到的蜜源含量进行非支配位置等级排序和拥挤度大小计算, 并将所有非支配等级排序为 1 的位置作为非支配精英位置集合的初始位置。

Step 05 根据文中所述方法分别在基础膜和表层膜内获得每只蜜蜂的局部最优位置、每个膜内的全局最优位置和所挑选工蜂的局部最优位置, 并根据式 (6.3.4) ~ 式 (6.3.7) 对基础膜和表层膜内蜜蜂的量子位置进行更新, 并根据式 (6.3.12) 获得食物源位置。

Step 06 根据 **Step 03** 中的方法对所有蜜蜂更新后的位置进行调整, 得到满足频谱分配的可行解, 并计算每个位置的蜜源含量。对基础膜内每只蜜蜂的局部最优位置、各个基础膜的膜内全局最优位置及表层膜的单目标最优位置集合进行更新, 并把表层膜内所有蜜蜂更新后的位置存入非支配精英位置集合。

Step 07 在每个膜内的算法演进过程中, 若当前迭代次数 t 满足 $\text{mod}(t, N_g) = 0$ ($\text{mod}(\cdot)$ 为取余数函数), 则执行通信规则, 即每个膜内的算法每连续迭代 N_g 次时, 表层膜和基础膜使用通信规则进行信息传递: 第 r 个基础膜将其膜内第 r ($r=1,2$) 个目标的全局最优位置 \mathbf{g}_r^{t+1} 和表层膜内单目标最优位置集合中的 \mathbf{b}_r^{t+1} 相比较, 选出较优秀的位置, 用于更新基础膜的膜内全局最优位置和表层膜的

单目标最优位置集合，同时每个基础膜将其膜内的全局最优位置传送并保存到表层膜的非支配精英位置集合；否则，直接转入 Step 08。

Step 08 获得添加新位置的非支配精英位置集合后，对集合内的位置进行非支配位置等级排序和拥挤度计算。对于非支配位置排序等级相同的那些位置，则需要进一步按照拥挤度值进行降序排列，并从中选出非支配等级较高且拥挤度较大的前 h_e 个位置作为新的非支配精英位置集合。

Step 09 若没有达到最大迭代次数，则需要设置 $t = t + 1$ ，并返回 Step 05；否则，将非支配精英位置集合中非支配位置排序等级为 1 的解输出，即为最终的非支配解集。每个基础膜将其膜内的全局最优位置传送到表层膜，对单目标最优位置集合进行更新并从表层膜输出，算法终止。

6.3.4 频谱分配实验仿真

认知无线电频谱分配模型中的 B 、 L 和 C 矩阵的生成方法与文献[22]的附录所使用的方法一致。参考文献[6]、[7]和[8]，对遗传算法（GA）、量子遗传算法（QGA）和粒子群优化（PSO）的仿真参数进行设置。参考文献[22]设置敏感图论着色（CSGC）的仿真参数，使用非合作式标号的标号形式。

MQBCO 参数设置：每个膜内的蜜蜂数目为 20，基础膜参数设置为 $e_1 = 0.06$ ， $e_2 = 0.03$ ， $e_3 = 0.06$ ， $e_4 = 0.03$ ， $e_5 = 0.01$ ， $c_1 = c_2 = 0$ ；表层膜参数设置为 $e_1 = 0.06$ ， $e_2 = 0.03$ ， $e_3 = 0.06$ ， $e_4 = 0.03$ ， $e_5 = 0.01$ ， $c_1 = c_2 = 1/D$ ， $N_g = 10$ ， $h_e = 20$ 。GA、QGA 和 PSO 中的种群规模都设为 20。4 种算法的迭代次数均为 1000 次。

1. 单目标频谱分配实验

利用所提出的 MQBCO 对单目标频谱分配进行仿真时，设可用频谱的数目 $M=13$ ，认知用户的数目 $N=13$ ，授权用户的数目为 13。仿真结果为 200 次实验结果的平均值。如图 6.3.1 和图 6.3.2 所示分别为 5 种频谱分配方法所获得的最

大和网络效益和最大比例公平网络效益的平均值的收敛曲线。从图中可以观察到,所提出的 MQBCO 无论是在收敛速度还是在收敛精度方面均优于 GA、QGA 和 PSO; 其相比于 CSGC 算法,收敛精度也高出很多。可见,基于 MQBCO 的单目标频谱分配方法在 5 种方法中是最优秀的,也是最可靠的。

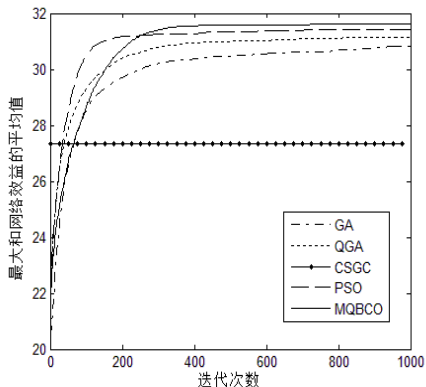


图 6.3.1 最大和网络效益和迭代次数关系曲线

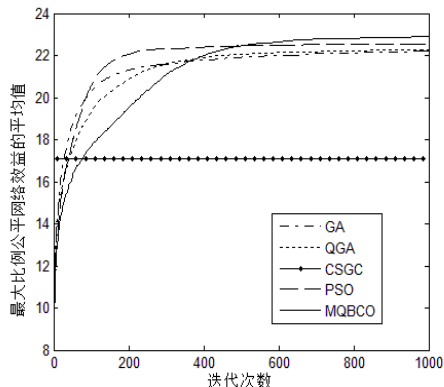


图 6.3.2 最大比例公平性网络效益和迭代次数关系曲线

将频谱池中的可用频谱数目设置为 $M=13$, 授权用户数目也设置为 13 时, 随着认知用户数目从 10 增加到 20, 5 种方法所得到的最大和网络效益随认知用户的变化情况如表 6.3.1 所示, 最大比例公平网络效益随认知用户数目的变化情况如表 6.3.2 所示。当认知用户数目增加时, 在可用信道确定的情况下, 认知用户之间及认知用户与授权用户之间产生的干扰将会增加, 并且认知用户之间对有限信道资源的竞争也会变得更为剧烈, 导致用户所获得的平均网络效益会不断减少。从表 6.3.1 和表 6.3.2 中不难发现, 将最大和网络效益、最大比例公平网络效益分别作为优化目标时, 基于 MQBCO 的频谱分配方法与其他 4 种方法相比, 能够获得更高的网络效益, 性能明显优于后者。而且, 使用 MQBCO 方法只需仿真 1 次便可求解出 2 种网络效益函数的最优解, 相比于其他方法仿真 1 次仅能求解出 1 种网络效益函数的近似最优解, 所提方法的高效性得以验证。

表 6.3.1 最大和网络效益比较

算法	认知用户数目					
	10	12	14	16	18	20
CSGC	36.806	30.870	27.498	25.280	22.904	20.772
GA	38.505	33.834	28.210	26.091	24.530	21.744
QGA	38.761	34.134	28.514	26.286	24.656	21.817
PSO	41.153	35.695	32.455	29.302	26.633	24.708
MQBCO	41.279	35.842	32.576	29.522	26.784	24.847

表 6.3.2 最大比例公平网络效益比较

算法	认知用户数目					
	10	12	14	16	18	20
CSGC	24.548	19.266	12.839	8.8101	5.8347	4.3041
GA	27.976	24.094	20.837	17.530	15.379	14.248
QGA	28.110	24.281	20.764	17.420	15.217	13.871
PSO	28.275	24.519	21.045	17.782	15.662	14.397
MQBCO	28.759	24.942	21.475	17.904	15.801	14.488

2. 多目标频谱分配实验

仿真参数设置：授权用户数目设置为 20，认知用户数目设置为 $N=10$ ，可用频谱数目设置为 $M=10$ ，两种多目标情况下的频谱分配仿真结果分别如图 6.3.3 和图 6.3.4 所示。图中 MSR、MPF 和 MMR 分别表示最大和网络效益、最大比例公平网络效益和最大贫困用户网络效益，则 CSGC-MSR、CSGC-MPF 和 CSGC-MSR 分别表示使用 CSGC 对 MSR、MPF 和 MMR 进行单目标求解，PSO-MSR、PSO-MPF 和 PSO-MMR 分别表示使用 PSO 对 MSR、MPF 和 MMR 进行单目标求解。仿真过程中， B 、 L 和 C 矩阵对于每种方法均为相同的。图 6.3.4 的公平性使用 MPF 目标函数，在使用 MQBCO 获得的 Pareto 前端解集中，存在 1 个非支配解同时支配利用 CSGC 和 PSO 对目标 MPF 所优化得到的解，也存在 1 个非支配解同时支配用 CSGC 和 PSO 对目标 MSR 所优化得到的解。图 6.3.4 的公平性使用 MMR 目标函数，在使用 MQBCO 获得的 Pareto 前

端解集中, 存在 3 个非支配解同时支配 CSGC 和 PSO 对目标 MMR 所优化得到的解, 也存在 1 个非支配解同时支配 CSGC 和 PSO 对目标 MSR 所优化得到的解。不难发现, PSO 和 CSGC 单目标优化算法求出的解均被 MQBCO 的非支配解集中的解所支配。

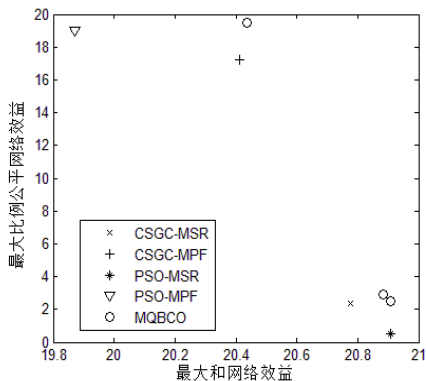


图 6.3.3 当 $N=M=10$ 时目标 MSR 和 MPF 的仿真结果

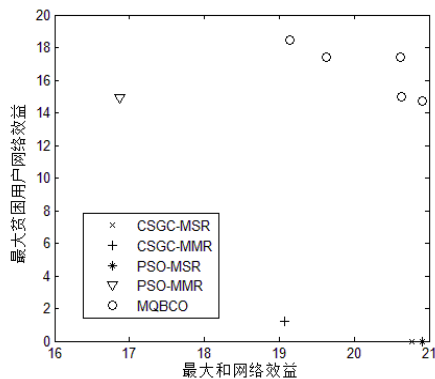


图 6.3.4 当 $N=M=10$ 时目标 MSR 和 MMR 的仿真结果

仿真参数设置: 授权用户数目设置为 20, 认知用户数目设置为 $N=10$, 可用频谱数目设置为 $M=20$, 两种多目标频谱分配的仿真结果如图 6.3.5 和图 6.3.6 所示。同样, 仿真过程中使用的 B 、 L 和 C 矩阵相同。从图 6.3.5 中可以观察到, MQBCO 的 Pareto 前端解集中有 1 个非支配解同时支配着用 CSGC 和 PSO 对目标 MSR 所优化得到的解, 也有 1 个解同时支配着用 CSGC 和 PSO 对目标 MPF 所优化得到的解。从图 6.3.6 中可以发现, 在使用 MQBCO 获得的 Pareto 前端解集中, 有 9 个解同时支配着用 CSGC 和 PSO 对目标 MMR 所优化得到的解, 也有 1 个解同时支配着用 CSGC 和 PSO 对目标 MSR 所优化得到的解。仿真实验表明, 所提出的基于 MQBCO 的多目标频谱分配方法所获得的网络效益均超过基于 PSO 和 CSGC 的频谱分配方法, 是一种高效可靠的频谱分配方法。

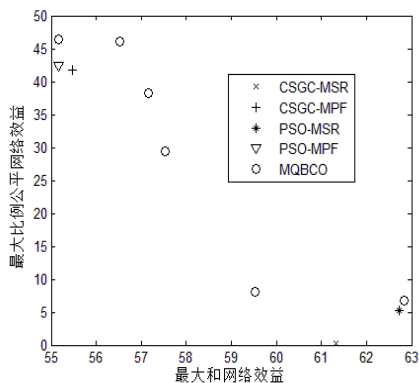


图 6.3.5 当 $N=10, M=20$ 时目标 MSR 和 MPF 的仿真结果

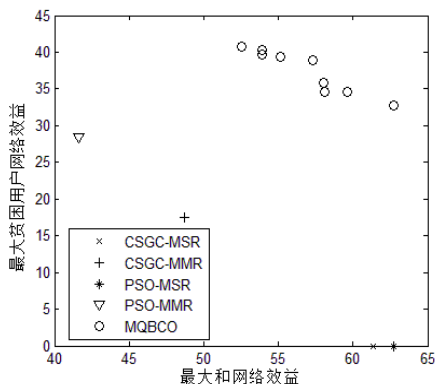


图 6.3.6 当 $N=10, M=20$ 时目标 MSR 和 MMR 的仿真结果

6.4 小结

为了获得认知无线电系统的分配效果较优的单目标频谱分配方案，本章首先提出了解决单目标频谱分配问题的量子粒子群优化算法，并通过仿真实验验证了所提方法明显优于现有的频谱分配方法，如基于粒子群算法、遗传算法、量子遗传算法的频谱分配及经典的颜色敏感图论着色方法。其次，为了解决多目标频谱分配问题，提出了基于膜量子蜂群优化的多目标频谱分配方法。其中所设计的多目标膜量子蜂群算法引入了人工蜂群算法的思想，并采用非支配解排序和拥挤度计算过程，可获得均匀分布的 Pareto 前端解。将 MQBCO 应用于多目标频谱分配问题中，可有效兼顾认知无线电系统的总体网络效益和用户间的公平性，是一种高效可靠的频谱分配方法。因此可以得出结论，所提出的多目标膜量子蜂群算法是一种高效可靠的算法，也可用于解决其他复杂的多目标优化问题。

参考文献

- [1] 李一兵, 杨蕊, 高振国. 基于着色理论的认知无线电频谱分配算法. 系统工程与电子技术, 2010, 32 (6): 1109-1112.
- [2] Clancy T C. Dynamic spectrum access using the interference temperature model. Annales des Telecommunications/Annals of Telecommunications, 2009, 64(7-8):573-592.
- [3] Huang J, Berry R, Honig M L. Auction-based spectrum sharing. ACM Mobile Networks and Applications (MONET), 2006, 11(3):405-418.
- [4] Niyato D, Hossain E. Competitive spectrum sharing in cognitive radio networks: A dynamic game approach. IEEE Transactions on Wireless Communications, 2008,7(7):2651-2660.
- [5] 郭彩丽, 冯春燕, 曾志民. 认知无线网络技术及应用. 北京: 电子工业出版社, 2010.
- [6] Zhao Z, Peng Z, Zheng S, et al. Cognitive radio spectrum allocation using evolutionary algorithms. IEEE Transactions on Wireless Communications, 2009, 8(9): 4421-4425.
- [7] 赵知劲, 彭振, 郑仕链, 等. 基于量子遗传算法的认知无线电频谱分配. 物理学报, 2009, 58 (2): 1358-1363.
- [8] 赵知劲, 徐世宇, 郑仕链等. 基于二进制粒子群算法的认知无线电决策引擎. 物理学报, 2009, 58 (7): 5118-5125.
- [9] Hongyuan Gao, Jinlong Cao, Yuning Zhao. Membrane quantum particle swarm optimisation for cognitive radio spectrum allocation. Int. J. Computer Applications in Technology, 2012: 43(4):359-265.

- [10] Hongyuan Gao, Jinlong Cao. Membrane-inspired quantum shuffled frog leaping algorithm for spectrum allocation. *Journal of Systems Engineering and Electronics*, 2012,23(5):679-688.
- [11] 高洪元, 曹金龙. 认知无线电中的量子蛙跳频谱分配. *应用科学学报*, 2014, 32 (1): 19-26.
- [12] 刁鸣, 张志强, 高洪元. 离散量子粒子群优化的认知无线电频谱分配. *计算机工程*, 2015, 41 (11): 126-130.
- [13] 李晨琬. 基于量子群智能的认知无线电关键技术研究. 哈尔滨工程大学硕士学位论文, 2015.
- [14] Zheng H, Peng C. Collaboration and fairness in spectrum assignment in opportunistic spectrum access. 40th IEEE International Conference on Communications(ICC), 2005: 3132-3136.
- [15] Gao Hongyuan, Cao Jinlong, Diao Ming. A simple quantum-inspired particle swarm optimization and its applicationm. *Information Technology Journal*, 2011, 10(12): 2315-2321.
- [16] 高洪元, 李晨琬. 膜量子蜂群优化的多目标频谱分配. *物理学报*, 2014, 63 (12): 128802-128802.
- [17] Bora T C, Lebensztajn L, Coelho L D S. Non-dominated sorting genetic algorithm based on reinforcement learning to optimization of broad-band reflector antennas satellite. *IEEE Transactions onMagnetics*, 2012, 48(2): 767-770.
- [18] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.
- [19] 高洪元, 曹金龙. 量子蜂群算法及其在认知频谱分配中的应用. *中南大学学报 (自然科学版)*, 2012, 43 (12): 4744-4749.

- [20] Gao Hongyuan, Cao Jinlong. Membrane-inspired quantum shuffled frog leaping algorithm for spectrum allocation. *Journal of Systems Engineering and Electronics*, 2012, 23(5): 679-688.
- [21] Gao Hongyuan, Cao Jinlong. Non-dominated sorting quantum particle swarm optimization and its application in cognitive radio spectrum allocation. *Journal of Central South University*. 2013, 20(7):1878-1888.
- [22] Peng C, Zheng H, Zhao B Y. Utilization and fairness inspectrum assignment for opportunistic spectrum access. *Mobile and Networking Technologies for Social Applications*, 2006, 11(4):555-576.

量子群智能的频谱感知技术

频谱感知作为认知无线电的一项关键技术，它的主要功能在于检测可供认知用户使用的频谱空穴，同时监测主用户信号的活动情况，保证主用户再次使用频谱时，认知用户能快速退出相应频段。目前已提出的频谱感知方法主要包括匹配滤波器检测、能量检测、周期平稳特征检测及多分辨率频谱感知^[1]，这些频谱感知方法均为单节点感知方法。然而，在阴影和深度衰落情况下，单个节点的感知结果并不可靠。因此，需要对多个节点的感知结果进行融合，以提高其检测的可靠性，即协作频谱感知技术^[2]。文献[3]改进了一种线性协作感知框架，但如何在最短的时间内求解出最优权向量是一个优化难题。

为了克服无线通信阴影和衰落的影响，在单节点检测的基础上提出了协作频谱感知方法。它利用一定的融合算法，对得到的多个认知用户的频谱感知结果、检验统计量或数据本身进行融合处理，并在此基础上对频谱的使用情况进行估计判断^[4]。

协作频谱感知主要包括认知用户域和融合中心^[5,6]。认知用户域是为了实现各个认知用户的合作，如参与的认知用户数量、认知用户之间的通信方式、认知用户的合作方式及频谱感知结果的上报策略。融合中心主要针对集中式的频

谱感知算法而言，它的主要作用是根据认知用户域上报的频谱结果按照一定的融合算法对其进行融合，并且最终给出频谱的利用情况。

改进群算法、神经网络、猫群算法和混合蛙跳算法等连续优化算法都可以用于频谱感知^[7-10]。但大量的仿真实验表明，因为一些简单的智能优化算法会陷入局部收敛，所以在认知用户数较大的情况下很难获得最优的仿真结果^[11,12]。

本章将所设计的量子细菌觅食算法应用于解决认知无线电协作频谱感知问题，并进行了计算机仿真。仿真结果表明，基于量子细菌觅食算法的频谱感知方法相对于经典的智能计算频谱感知方法而言，具有很大优势。

7.1 线性协作频谱感知模型

频谱感知技术需要可靠地检测到各种形式的认知用户的微弱信号，才能有效地检测到频谱空穴和监测主用户信号的活动情况。一些单点检测方法由于易受到阴影和深度衰落^[13]的影响而使得结果并不可靠。为了解决上述问题，可以使用线性协作频谱感知模型来构建认知用户之间的合作检测模式。线性协作频谱感知模型如图 7.1.1 所示。假设认知网络中有 M 个认知用户相互协作进行协作感知，时刻 k 时各个认知用户的本地感知二元假设检验模型如下：

$$\begin{cases} H_0 : x_j(k) = v_j(k) \\ H_1 : x_j(k) = h_j s(k) + v_j(k) \end{cases} \quad (7.1.1)$$

其中， $j=1,2,\dots,M$ ， $s(k)$ 为主用户发射的信号，认知网络中的每一个认知用户都可以接收到该信号。信号受到信道衰减的影响，信道衰减为 h_j ，在整个检测过程中认为信道衰减始终是常数。 $x_j(k)$ 为第 j 个认知用户接收到的信号，其受到加性高斯白噪声 $v_j(k)$ 的影响，方差用向量 $\sigma=[\sigma_1^2,\sigma_2^2,\dots,\sigma_M^2]^T$ 表示。

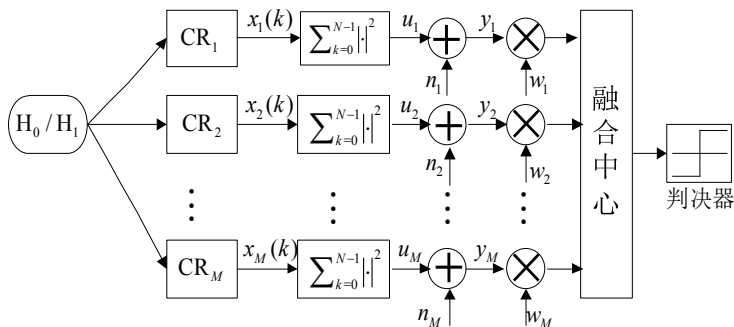


图 7.1.1 线性协作频谱感知框图

各个认知用户的本地感知采用能量检测的方法，在采样间隔内经过 N 点采样，所计算的判决统计量为

$$u_j = \sum_{k=0}^{N-1} |x_j(k)|^2 \quad (7.1.2)$$

随后统计量经过控制信道传输到融合中心，融合中心收到的各个统计量表示为

$$y_j = u_j + n_j \quad (7.1.3)$$

其中 n_j 为控制信道引入的噪声，假设其服从均值为 0 的高斯分布，方差用向量 $\boldsymbol{\delta} = [\delta_1^2, \delta_2^2, \dots, \delta_M^2]^T$ 表示。融合中心根据收到的每个认知用户的信号 y_j ，计算出的全局判决统计量为

$$y_c = \sum_{j=1}^M w_j y_j \quad (7.1.4)$$

其中， $\boldsymbol{w} = [w_1, w_2, \dots, w_M]$ 为控制判决的权重向量，权重向量反映了特定的认知用户对全局检测的贡献大小。例如，如果某一认知用户接收到的信号的信噪比很高，则它得到的检测统计量更能反映实际情况，因此给它分配较高的权重系数。而如果某一认知用户经受了深度阴影和衰落，则给它分配低的权重系数，以降低它对融合判决结果的影响。

融合中心将 y_c 与特定门限 γ_c 进行比较，实现最终判决。如果 $y_c \geq \gamma_c$ ，则认为授权用户信号存在；否则，认为授权用户信号不存在。在这种协作频谱感

知模型中，虚警概率的表达式为

$$P_f = Q\left[\frac{\gamma_c - N\boldsymbol{\sigma}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T}}\right] \quad (7.1.5)$$

检测概率的表达式为

$$P_d = Q\left[\frac{\gamma_c - (N\boldsymbol{\sigma} + E_s \mathbf{h})^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}}\right] \quad (7.1.6)$$

其中， $Q(x) = \int_x^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$ ， $E_s = \sum_{k=0}^{N-1} |s(k)|^2$ ， $\mathbf{A} = 2N\text{diag}^2(\boldsymbol{\sigma}) + \text{diag}(\boldsymbol{\delta})$ ， $\mathbf{B} = 2N\text{diag}^2(\boldsymbol{\sigma}) + \text{diag}(\boldsymbol{\delta}) + 4E_s \text{diag}(\mathbf{h}) \text{diag}(\boldsymbol{\sigma})$ ， $\mathbf{h} = [|h_1|^2, |h_2|^2, \dots, |h_M|^2]^T$ 。

认知无线电的目的就是在给定虚警概率 P_f 的情况下，想要获得最大的检测概率。门限 γ_c 用虚警概率表示为

$$\gamma_c = N\boldsymbol{\sigma}^T \mathbf{w}^T + Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} \quad (7.1.7)$$

将式 (7.1.7) 代入式 (7.1.6)，得到

$$P_d = Q\left[\frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}}\right] \quad (7.1.8)$$

由于 Q 函数是单调递减函数，因此实现上式最大化就相当于下式最小化，定义

$$f(\mathbf{w}) = \frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}} \quad (7.1.9)$$

7.2 合作式频谱感知的基本算法

7.2.1 基于修正偏差因子的频谱感知方法

文献[14]在提出了一种结构简单的线性协作频谱感知模型的同时，也给出

了一种解决该协作感知问题的方法——修正偏差因子（Modified Deflection Coefficient, MDC）方法。由于直接获得式（7.1.9）的解析解比较困难，修正偏差因子方法并不是直接优化式（7.1.9）来解决该优化问题，而是将式（7.1.9）的最小化优化问题转化为对 MDC 的最大化优化。MDC 的表达式如下：

$$d_m^2(\mathbf{w}) = \frac{[E(y_c | H_1) - E(y_c | H_0)]^2}{\text{Var}(y_c | H_1)} = \frac{(E_s \mathbf{h}^T \mathbf{w}^T)^2}{\mathbf{w} \mathbf{B} \mathbf{w}^T} \quad (7.2.1)$$

其中， $E(*)$ 表示求均值， $\text{Var}(*)$ 表示求方差。

MDC 方法用 $d_m^2(\mathbf{w})$ 来度量认知协作频谱感知问题中的检测概率，上述问题便转化为对 $d_m^2(\mathbf{w})$ 的最大化优化问题。

$$\max_{\mathbf{w}} d_m^2(\mathbf{w}) \quad \text{st. } \|\mathbf{w}\|_2^2 = 1 \quad (7.2.2)$$

由此求得的一个最优权重向量为

$$\mathbf{w}^* = \frac{\mathbf{B}^{-\frac{1}{2}} \mathbf{B}^{-\frac{1}{2}} \mathbf{h}}{\|\mathbf{B}^{-\frac{1}{2}} \mathbf{B}^{-\frac{1}{2}} \mathbf{h}\|_2} \quad (7.2.3)$$

7.2.2 基于粒子群算法的频谱感知

在粒子群算法中，每一个最优候选解被称为一个“粒子”，若干个粒子就构成了粒子的群体。每个粒子没有重量和体积，通过目标函数确定它的适应度值。每个粒子在解的空间中运动，并由速度决定它的运动方向和距离，粒子通过追随自身的个体最好位置与群体的全局最好位置来动态地调整自己的位置信息。

现考虑最小化优化问题，假设搜索空间为 M 维，粒子群规模为 S ，则在第 t 次迭代中，第 i 个粒子的当前位置为 $\mathbf{X}_i(t) = [X_{i1}(t), X_{i2}(t), \dots, X_{iM}(t)]$ ，第 i 个粒子的当前速度为 $\mathbf{V}_i(t) = [V_{i1}(t), V_{i2}(t), \dots, V_{iM}(t)]$ ，第 i 个粒子的当前个体最好位置为 $\mathbf{P}_i(t) = [P_{i1}(t), P_{i2}(t), \dots, P_{iM}(t)]$ ， $1 \leq i \leq S$ ，其更新公式为

$$\mathbf{P}_i(t) = \begin{cases} \mathbf{X}_i(t), & f[\mathbf{X}_i(t)] < f[\mathbf{P}_i(t-1)] \\ \mathbf{P}_i(t-1), & f[\mathbf{X}_i(t)] \geq f[\mathbf{P}_i(t-1)] \end{cases} \quad (7.2.4)$$

整个粒子群搜索到的当前最好位置称为全局最好位置，表示为 $G(t) = [G_1(t), G_2(t), \dots, G_M(t)]$ 。基本粒子群算法的速度和位置更新公式如下：

$$V_{ij}(t+1) = V_{ij}(t) + c_1 \cdot r_1 \cdot (P_{ij}(t) - X_{ij}(t)) + c_2 \cdot r_2 \cdot (G_j(t) - X_{ij}(t)) \quad (7.2.5)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (7.2.6)$$

其中， $1 \leq i \leq S$ ， $1 \leq j \leq M$ ； t 表示第 t 次迭代； c_1 和 c_2 称为学习因子，一般取 2， c_1 表示调节粒子飞向自身最好位置方向的步长， c_2 表示调节粒子飞向全局最好位置方向的步长； r_1 和 r_2 都为 $[0,1]$ 之间均匀分布的随机数。为了减少粒子在搜索过程中飞离搜索空间的概率，可将粒子的速度变化限制在一定的范围内 $V_{ij}(t) \in [-V_j^{\max}, V_j^{\max}]$ ，将粒子的位置限定在搜索空间内 $X_{ij}(t) \in [-X_j^{\max}, X_j^{\max}]$ 。

通过以上对于粒子群算法的介绍，将基于粒子群算法的协作频谱感知方法的流程归纳如下。

Step 01 参数初始化。设置 $t=0$ ，设定粒子群体规模为 S 。对任意 ij ，随机产生初始位置和初始速度 $X_{ij}(0) \in [-X_j^{\max}, X_j^{\max}]$ ， $V_{ij}(0) \in [-V_j^{\max}, V_j^{\max}]$ 。初始化 $P_{ij}(0) = X_{ij}(0)$ 。对每个粒子的初始位置进行归一化处理。

Step 02 计算每个粒子当前位置 $X_i(t)$ 的适应度值 $f[X_i(t)]$ 。

Step 03 对于每个粒子，将 $X_i(t)$ 的适应度值 $f[X_i(t)]$ 与其个体最好位置 $P_i(t-1)$ 的适应度值 $f[P_i(t-1)]$ 进行比较，若优于 $f[P_i(t-1)]$ ；则令 $P_i(t) = X_i(t)$ ；否则， $P_i(t) = P_i(t-1)$ 。

Step 04 对于每个粒子，将 $X_i(t)$ 的适应度值 $f[X_i(t)]$ 与粒子群所经历的全局最好位置 $G(t-1)$ 的适应度值 $f[G(t-1)]$ 进行比较，若优于 $f[G(t-1)]$ ；则令 $G(t) = X_i(t)$ ；否则， $G(t) = G(t-1)$ 。

Step 05 根据式 (7.2.5) 和 (7.2.6)，对粒子的速度和位置进行更新。对新的速度和位置进行钳位处理，并对经过钳位处理的粒子位置进行归一化处理。

Step 06 如果未达到结束条件（达到一个预先设定的最大迭代次数），则设置 $t=t+1$ ，进入 **Step 02**；否则，终止迭代，输出最优解。

7.2.3 计算机仿真

本节将基于修正偏差因子的协作感知方法与基于经典粒子群算法的协作感知方法进行比较, 仿真过程中分别选取虚警概率 $P_f=0.1$ 、 $P_f=0.2$ 和 $P_f=0.3$, 仿真结果列在表 7.2.1 和表 7.2.2 中。假设授权用户发出的信号 $s(k)=1$, 采样次数 $N=20$, 仿真过程中进行 200 次独立试验。仿真参数设置如下: $\sigma=[1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0]^T$; $\delta=[1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0]^T$; $h=[0.6,0.5,0.7,0.3,0.4,0.5,0.3,0.4]^T$ 。粒子群算法的参数设置为: 学习因子 $c_1=c_2=2$, 粒子总数为 50, 全部粒子的速度变化范围限定为定义区间的 10%, 算法迭代次数为 100 次。从仿真结果可以看出, 基于粒子群算法的感知方法要明显优于基于修正偏差因子的感知方法, 从而也证实了修正偏差因子方法确实不是一种全局最优的求解方法。

表 7.2.1 有 8 个本地认知用户时的目标函数仿真结果

虚警概率	$P_f=0.1$	$P_f=0.2$	$P_f=0.3$
MDC	-2.08374	-2.39648	-2.62200
PSO	-2.08581	-2.39720	-2.62202

表 7.2.2 有 8 个本地认知用户时的检测概率仿真结果

虚警概率	$P_f=0.1$	$P_f=0.2$	$P_f=0.3$
MDC	0.98140	0.99172	0.99562
PSO	0.98150	0.99173	0.99563

表 7.2.3 和表 7.2.4 给出了另一种仿真参数条件下的仿真结果。仿真参数设置: $\sigma=[0.9,1.3,1.0,2.0,0.8,1.8]^T$; $\delta=[0.4,0.6,0.4,0.2,0.5,0.4]^T$; $h=[0.6,0.5,0.7,0.3,0.4,0.5]^T$ 。其余参数设置与上一仿真试验相同。从仿真结果可以进一步证实修正偏差因子方法不是一种全局最优的求解方法。

表 7.2.3 有 6 个本地认知用户时的目标函数仿真结果

虚警概率	$P_f=0.1$	$P_f=0.2$	$P_f=0.3$
MDC	-1.69131	-1.99700	-2.21742
PSO	-1.69539	-1.99867	-2.21797

表 7.2.4 有 6 个本地认知用户时的检测概率仿真结果

虚警概率	$P_f=0.1$	$P_f=0.2$	$P_f=0.3$
MDC	0.95461	0.97708	0.98670
PSO	0.95499	0.97717	0.98672

7.3 基于连续量子细菌觅食算法的频谱感知技术

7.3.1 量子细菌觅食算法的频谱感知^[15]

采用量子细菌觅食算法优化式 (7.1.9)，就可以将协作频谱感知检测概率最大化问题转化为求如下优化函数

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}} \quad (7.3.1)$$

其最优解由权重向量 \mathbf{w}^* 决定，该权重向量同样使得在给定虚警概率 P_f 下取得最大的检测概率。另外，如果权重向量 \mathbf{w}^* 是使得式 (7.3.1) 最小化的最优解，则 $\lambda \mathbf{w}^*$ 也将是使得式 (7.3.1) 最小化的最优解，最优解的个数将有无穷多个。因此，对权重向量做如下的归一化的约束可以得到唯一的最优解。

$$\min_{\mathbf{w}} J(\mathbf{w}), \text{ s.t. } 0 \leq w_j \leq 1, j=1, 2, \dots, M, \sum_{j=1}^M w_j = 1 \quad (7.3.2)$$

由此设计适应度函数为

$$\begin{cases} J(\mathbf{w}) = \frac{Q^{-1}(P_f) \sqrt{\mathbf{w} \mathbf{A} \mathbf{w}^T} - E_s \mathbf{h}^T \mathbf{w}^T}{\sqrt{\mathbf{w} \mathbf{B} \mathbf{w}^T}} \\ \text{s.t. } 0 \leq w_j \leq 1, j=1, 2, \dots, M, \sum_{j=1}^M w_j = 1 \end{cases} \quad (7.3.3)$$

此优化问题的最优解就是使得式 (7.3.3) 最小化的最优权重向量。

在基于量子细菌觅食算法的协作感知方法中，设计了基于量子细菌觅食算

法的融合中心，其原理框图如图 7.3.1 所示。对于协作频谱感知模型，量子细菌的位置与量子位置是等价的，在量子域[0,1]内随机产生量子位。因此，对于协作感知模型的目标函数的优化问题可以等价于量子细菌觅食算法的适应度函数的优化问题，第 i 个量子细菌的位置 $\bar{\theta}_i = [\bar{\theta}_{i1}, \bar{\theta}_{i2}, \dots, \bar{\theta}_{iM}]$ 与该细菌的归一化的量子位置 $\theta_i = [\theta_{i1}, \theta_{i2}, \dots, \theta_{iM}]$ 等价，它的适应度函数可以表示为

$$\begin{cases} J(\bar{\theta}_i) = \frac{Q^{-1}(P_f) \sqrt{\bar{\theta}_i A(\bar{\theta}_i)^T} - E_s \mathbf{h}^T(\bar{\theta}_i)^T}{\sqrt{\bar{\theta}_i B(\bar{\theta}_i)^T}} \\ \text{s.t. } 0 \leq \bar{\theta}_{id} \leq 1, d=1, 2, \dots, M, \sum_{d=1}^M \bar{\theta}_{id} = 1 \end{cases} \quad (7.3.4)$$

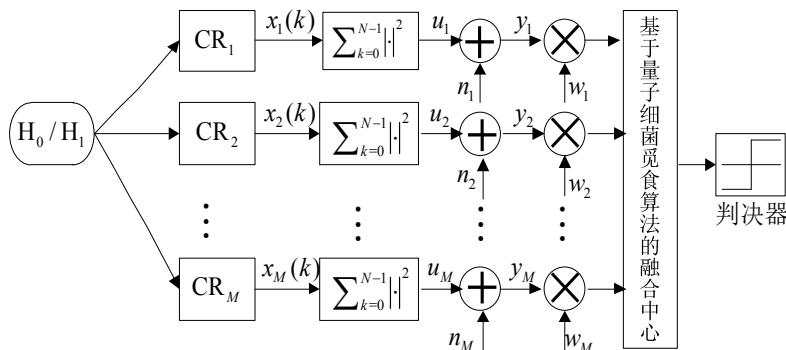


图 7.3.1 基于量子细菌觅食算法的协作频谱感知模型框图

综上所述，所提出的基于量子细菌觅食算法的协作频谱感知权重向量算法的流程如下。

Step 01 初始化参数 S 、 N_c 、 N_s 、 N_{re} 、 N_{ed} 、 P_{ed} 和 θ_i ，其中， $i=1, 2, \dots, S$ 。初始化的量子位位置在量子域内随机产生，对初始化的量子位置进行归一化处理。

Step 02 迁徙操作循环： $l=l+1$ ，令 $k=0$ 。

Step 03 复制操作循环： $k=k+1$ ，令 $j=0$ 。

Step 04 趋向性操作循环： $j=j+1$ 。

[a] 保存当前量子细菌的量子位置 θ_i^j ，其中， $i=1, 2, \dots, S$ 。

[b] 寻找当前的最优适应度值, 保存其对应最优量子细菌的量子位置 $\theta_{\min}^j = [\theta_{\min 1}^j, \theta_{\min 2}^j, \dots, \theta_{\min M}^j]$ 。

[c] 对于每个量子细菌, 按照如下方式进行一次趋向性操作。

[d] 计算适应度函数 $J^i(j, k, l)$ 。

[e] 令 $J_{last}^i = J^i(j, k, l)$ 。

[f] 旋转: 计算全部 S 个量子细菌的量子位置的平均值 $z^j = [z_1^j, z_2^j, \dots, z_M^j]$ 。

[g] 移动: 根据公式计算新的量子位置 $\phi_{id}^{j+1} = r_1 \cdot (\theta_{id}^j - \theta_{\min d}^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j)$, $\theta_{id}^{j+1} = |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}|$, 其中 $d=1, 2, \dots, M$ 。对第 $j+1$ 代趋向性操作的量子细菌的量子位置 $\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 进行归一化处理。

[h] 用 $\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 的映射位置计算适应度值 $J^i(j+1, k, l)$ 。

[i] 游弋:

(1) 令 $m=0$ 。

(2) while $m < N_s$ 时:

1) 令 $m=m+1$ 。

2) 如果 $J^i(j+1, k, l) < J_{last}^i$, 保存 $J_{last}^i = J^i(j+1, k, l)$, $\theta_i^j = \theta_i^{j+1}$, 根据公式计算 $\phi_{id}^{j+1} = r_1 \cdot (\theta_{id}^j - \theta_{\min d}^j) + \beta \cdot \text{sign}(r) \cdot \log(r_2) \cdot (\theta_{id}^j - z_d^j)$, $\theta_{id}^{j+1} = |\theta_{id}^j \cos \phi_{id}^{j+1} - \sqrt{1 - (\theta_{id}^j)^2} \sin \phi_{id}^{j+1}|$, 其中 $d=1, 2, \dots, M$ 。对第 $j+1$ 代趋向性操作的量子细菌的量子位置 $\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 进行归一化处理。用新的映射位置 $\theta_i^{j+1} = [\theta_{i1}^{j+1}, \theta_{i2}^{j+1}, \dots, \theta_{iM}^{j+1}]$ 计算 $J^i(j+1, k, l)$ 。

3) 否则, 因为 $m = N_s$, 结束 while 循环, 设置 $\theta_i^{j+1} = \theta_i^j$ 。

[j] 如果 $i \neq S$, 进入步骤[b]处理下一个量子细菌。

Step 05 如果 $j < N_c$, 则进入 **Step 04** 继续进行趋向性操作。

Step 06 复制操作:

[a] 在给定 k 和 l 的情况下, 对于每一个量子细菌 $i=1,2,\dots,S$, 使得健康值 $J_{\text{health}}^i = J^i(N_c, k, l)$ 表示第 i 个量子细菌的健康状况。按照健康值的降序对所有量子细菌进行排序。

[b] 具有较差健康值的 S_r 个量子细菌消亡, 保留的 S_r 个量子细菌分裂(该过程意味着新复制的子代量子细菌代替消亡的量子细菌位于与它们父代量子细菌相同的位置)。

Step 07 如果 $k < N_{re}$, 则进入 **Step 03**, 继续完成下一代复制操作, 开始进行新一轮的趋向性操作。

Step 08 迁徙操作: 对于每一个非最优量子细菌, 以概率 P_{ed} 进行变异。如果对某一量子细菌进行变异, 则在量子域内随机产生一个新的量子细菌。如果 $l < N_{ed}$, 则进入 **Step 02**; 否则终止算法。

为了计算与比较的需要, 令 ite 表示 BFOA 和 QBFOA 算法的当前迭代次数, $ite = N_c \cdot (k-1) + N_c \cdot N_{re} \cdot (l-1) + j$ 。

7.3.2 计算机仿真

假设具有 M 个认知用户的认知网络, 其中每个认知用户可以相互独立地检测目标频谱。认知网络中的权重向量 $\mathbf{w} = [w_1, w_2, \dots, w_M]$ 为一个 M 维向量, 在所提出的基于量子细菌觅食算法的协作感知模型中, 第 i 个量子细菌的维数与权重向量的维数相等, 也就与认知用户数相等, 由此完成了量子细菌与待优化权重向量之间的映射。为简单起见, 假设授权用户发出的信号 $s(k)=1$, 采样次数 $N=20$ 。

仿真过程中, 在协作频谱感知这个工程问题的背景下, 将量子细菌觅食算法 (QBFOA) 与传统蛙跳算法 (SFLA)、细菌觅食算法 (BFOA) 和粒子群算法 (PSO) 相比较。量子细菌觅食算法的主要参数设置如下: $S=50$, $N_c=100$, $N_s=2$, $N_{re}=5$, $N_{ed}=2$, $P_{ed}=0.1$, $\beta=1$ 。粒子群算法^[16]的参数设置为: 学习因子 $c_1 = c_2 = 2$, 粒子总数为 50, 全部粒子的速度变化范围限定为定义区间的 10%。

对于细菌觅食算法^[17], 参数与量子细菌觅食算法保持一致。另外, 单位游弋步长设置为 $c(i)=0.01$ 。

混合蛙跳算法^[18]的参数设置为: 青蛙群体个体总数为 50, 族群数为 10, 每个族群青蛙数为 5, 族群内最大迭代次数为 5, 最大演进代数数为 1000。另外 d_j^{\max} 和 d_j^{\min} 设置为: $d_j^{\max}=0.5$, $d_j^{\min}=-0.5$ 。

图 7.3.2 和图 7.3.3 分别给出了假设网络中有 10 个认知用户并且在给定虚警概率 $P_f=0.1$ 时, 最优目标函数值和检测概率随迭代次数的变化曲线。进行 200 次独立试验, 智能计算算法的终止迭代次数设为 200。其他与认知无线电协作频谱感知模型相关的各个参数设置如下: $\sigma=[2.0,2.5,0.9,2.7,1.3,3.3,2.0,2.5,0.9,2.7]^T$; $\delta=[1.3,0.8,2.0,3.8,2.3,0.4,1.3,0.8,2.0,3.1]^T$; $h=[0.4,0.5,0.7,0.3,0.4,0.3,0.6,0.5,0.2,0.3]^T$ 。

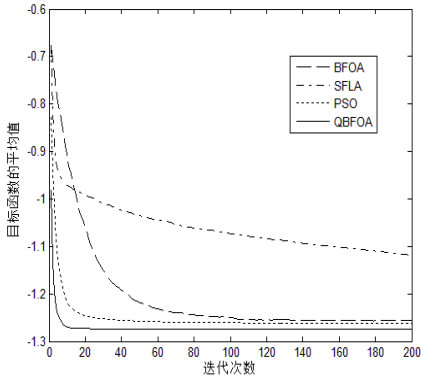


图 7.3.2 有 10 个本地认知用户时的目标函数仿真

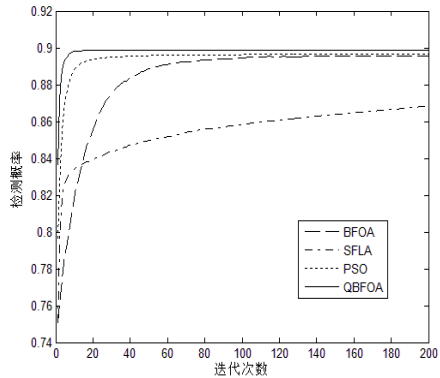


图 7.3.3 有 10 个本地认知用户时的检测概率仿真

图 7.3.4 和图 7.3.5 则分别给出了假设网络中有 25 个认知用户且给定虚警概率 $P_f=0.1$ 时, 最优目标函数值和检测概率随迭代次数的变化曲线。进行 200 次独立试验, 所有智能计算算法的终止迭代次数设为 400。参数设置如下: $\sigma=[2.0,2.5,0.9,2.7,1.3,3.3,2.0,2.5,0.9,2.7,2.1,2.3,0.7,2.8,1.1,3.6,2.1,2.3,1.9,2.2,2.0,2.7,1.1,3.2,1.5]^T$; $\delta=[1.3,0.8,2.0,3.8,2.3,0.4,1.3,0.8,2.0,3.1,1.1,0.6,2.1,3.5,2.5,0.3,1.5,0.7,2.2,3.3,1.4,0.6,2.2,3.6,2.2]^T$; $h=[0.4,0.5,0.7,0.3,0.4,0.3,0.6,0.5,0.2,0.3,0.3,0.4,0.4,0.5,0.3,$

0.4,0.5,0.6,0.1,0.5,0.3,0.7,0.6,0.5,0.2]^T。其余参数与上一仿真试验相同。

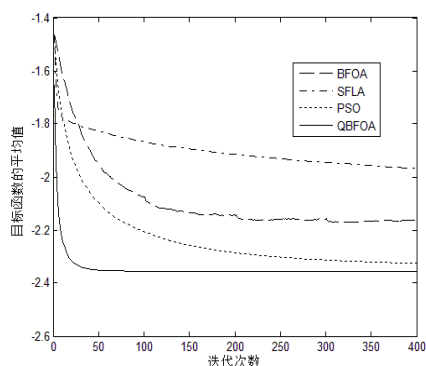


图 7.3.4 有 25 个本地认知用户时的
目标函数仿真

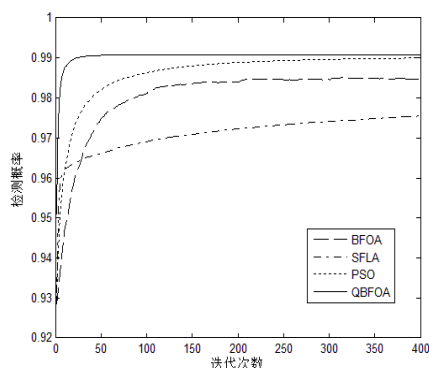


图 7.3.5 有 25 个本地认知用户时的
检测概率仿真

同时，本文还对另外一种低维情况进行了仿真，这时认知网络中只有 8 个本地认知用户。在这种情况下，分别使得 $P_f=0.05$ 、 $P_f=0.1$ 、 $P_f=0.15$ 和 $P_f=0.2$ 。参数 $\sigma=[2.0,2.5,0.9,2.7,1.3,3.3,2.0,2.5]^T$ ； $\delta=[1.3,0.8,2.0,3.8,2.3,0.4,1.3,0.8]^T$ ； $h=[0.4,0.5,0.7,0.3,0.4,0.3,0.6,0.5]^T$ 。仿真结果列在表 7.3.1 和表 7.3.2 中。从中可以看出，在所有 4 种算法之中，量子细菌觅食算法无论在收敛精度还是收敛速度方面都是最优的。

表 7.3.1 有 8 个本地认知用户时的目标函数仿真结果

算 法	$P_f=0.05$		$P_f=0.1$	
	平均值	标准差	平均值	标准差
BFOA	-0.9240	0.0026	-1.1813	0.0025
SFLA	-0.8393	0.0355	-1.1167	0.0260
PSO	-0.9298	0.0094	-1.1879	0.0070
QBFOA	-0.9319	0.0000	-1.1891	0.0000
算 法	$P_f=0.15$		$P_f=0.2$	
	平均值	标准差	平均值	标准差
BFOA	-1.3558	0.0026	-1.4958	0.0028
SFLA	-1.2845	0.0331	-1.4769	0.0138
PSO	-1.3635	0.0002	-1.5031	0.0002
QBFOA	-1.3639	0.0000	-1.5034	0.0000

表 7.3.2 有 8 个本地认知用户时的检测概率仿真结果

算 法	$P_f=0.05$		$P_f=0.1$	
	平均值	标准差	平均值	标准差
BFOA	0.8223	0.0007	0.8813	0.0005
SFLA	0.7993	0.0100	0.8679	0.0056
PSO	0.8238	0.0025	0.8826	0.0014
QBFOA	0.8243	0.0000	0.8828	0.0000
算 法	$P_f=0.15$		$P_f=0.2$	
	平均值	标准差	平均值	标准差
BFOA	0.9124	0.0004	0.9326	0.0004
SFLA	0.9005	0.0058	0.9302	0.0019
PSO	0.9136	0.0000	0.9336	0.0000
QBFOA	0.9137	0.0000	0.9337	0.0000

7.4 小结

本章把所设计的量子细菌觅食算法应用于认知无线电频谱感知问题中，构建出了基于量子细菌算法的协作感知方法，并且将其与经典的解决认知无线电频谱感知的算法进行比较，同时将它与传统蛙跳算法、细菌觅食算法和粒子群算法进行比较和仿真。仿真试验的结果证实了基于量子细菌觅食算法的频谱感知方法的优越性。

参考文献

- [1] Gu Junrong, Jang S J, Kim J M. A proactive dynamic spectrum access method against both erroneous spectrum sensing and asynchronous inter-channel spectrum sensing. KSII Transactions on Internet and Information Systems, 2012, 6(1):361-378.

- [2] Cabric D, Mishra S M, Brodersen R. Implementation issues in spectrum sensing for cognitive radios. In Proc. 38th Asilomar Conf. Signals, Systems and Computers, Pacific Grove, CA, United states, 2004: 772–776.
- [3] Guo J, T C Song, M Wu, D F Sun. An improved weighted cooperative spectrum sensing in cognitive radio networks. 3rd International Conference on CECNet, 2013:113-116.
- [4] Yücek T, Arslan H. A survey of spectrum sensing algorithms for cognitive radio applications. IEEE Communications Surveys and Tutorials, 2009, 11(1):116-130.
- [5] Chien W B, Yang C K, Huang Y H. Energy-saving cooperative spectrum sensing processor for cognitive radio system. IEEE Transactions on Circuits and Systems I: Regular Papers, 2011, 58(4):711-723.
- [6] Wang X Y, Wong A, Ho P H. Dynamically optimized spatiotemporal prioritization for spectrum sensing in cooperative cognitive radio. Wireless Networks, 2010, 16(4): 889-901.
- [7] 刁鸣, 钱荣鑫, 高洪元. 狼群优化的神经网络频谱感知算法. 计算机工程与应用, 2015.
- [8] 江辉, 陈飞飞, 杜文峰. 基于改进群搜索优化算法的认知无线电协作频谱感知. 电路与系统学报, 2013 (1): 187-192.
- [9] Pradhan P M, Panda G, Majhi B. Multiobjective cooperative spectrum sensing in cognitive radio using cat swarm optimization. Wireless Advanced (WiAd), 2012, 44-48.
- [10] 郑仕链, 楼才义, 杨小牛. 基于改进混合蛙跳算法的认知无线电协作频谱感知. 物理学报, 2010, 59 (5): 3611-3617.
- [11] 高洪元, 崔闻. 文化蛙跳算法及其在频谱感知中的应用. 中南大学学报, 2013, 44 (9): 3723-3730.

- [12] Gao Hongyuan, Cui Wen, Jiang Yilin. A quantum.-inspired shuffled frog leaping algorithm and its application in cognitive radio. *International Journal of Digital Content Technology and its Applications*, 2012, 20(6):32-42.
- [13] Ghasemi A, Sousa E. Collaborative spectrum sensing for opportunistic access in fading environments. In *Proc. IEEE Symp. New Frontiers in Dynamic Spectrum Access Networks*, Baltimore, MD, United states: Institute of Electrical and Electronics Engineers Computer Society, 2005:131-136.
- [14] Quan Z, Cui S, Sayed A H. Optimal Linear cooperation for spectrum sensing in cognitive radio networks. *IEEE journal of selected topics in signal processing*, 2008, 2(1): 28-40.
- [15] Gao Hongyuan, Cui Wen, Li Chenwan. A quantum bacterial foraging optimisation algorithm and its application in spectrum sensing. *International Journal of Modelling, Identification and Control*, 2013, 18(3): 234-242.
- [16] Kennedy J, Eberhart R. Particle swarm optimization. In *Proc. IEEE Int. Conf. Neural Netw.*, 1995: 1942-1948.
- [17] Passino K M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 2002, 22(3): 53-67.
- [18] Eusuff M, Lansey K. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 2003, 129(2): 210-225.

第 8 章

基于量子智能计算的 DOA 估计

波达方向 (Direction of Arrival, DOA) 估计通常又称空间谱估计或测向, 其在移动和卫星通信系统、信息战、雷达、被动声纳、地震学、射电天文学、导航、地球物理、声源跟踪、麦克风阵列、视频会议、时间序列分析、谱估计等方面有着广泛的应用, 引起了人们极大的研究兴趣^[1]。由于其应用的广泛性, 各种测向算法自提出之日起就得到了广泛关注, 40 多年来得到了长足的发展。空间谱估计要解决的基本问题是确定感兴趣空间信号的方位。1979 年, 美国的 R. O. Schmidt 等人提出了 MUSIC (Multiple Signal Classification) 算法^[2], 开创了子空间类阵列信号处理算法研究的先河, 实现了真正意义上的阵列高分辨测向, 也促进了特征子空间类算法的兴起。

子空间分解类算法从处理方式上分为两大类: 一类是以 MUSIC 为代表的噪声子空间类算法, 以 MUSIC 算法为代表的算法包括特征矢量法、MUSIC、求根 MUSIC 法及 MNM 等; 另一类是以旋转不变子空间为代表的信号子空间类算法 (Estimation of Signal Parameters via Rotational Invariance Techniques, ESPRIT)^[3], 该类方法利用信号子空间的旋转不变特性来估计信号方向, 避免了 MUSIC 算法因需要按量化间隔在整个搜索域搜索而带来的大量计算, 在一定程度上降低了算法对硬件的要求, 但算法的检测性能劣于 MUSIC 算法。以 ESPRIT 算法为代表的方法主要有 TAM、LS-ESPRIT 及 TLS-ESPRIT 等。

从 20 世纪 80 年代后期开始, 又出现了一类参数模型拟合算法, 其中比较有代表性的算法有最大似然 (Maximum Likelihood, ML) 算法^[4,5]和子空间拟合 (Subspace Fitting, SF) 算法^[6]等。ML 参数估计类方法包括确定性最大似然算法 (DML)^[4]和随机性最大似然算法 (SML)^[5]。由于 ML 参数估计方法的方向估计似然函数是非线性的, 所以求解最优解需进行多维搜索, 运算量巨大; 但与子空间分解类算法相比, 它的估计性能优良, 尤其是在低信噪比、小快拍数的情况下, 此类算法比 MUSIC 及其他子空间分解类算法性能好得多。另外, ML、SF 等参数模型拟合算法在相干信源情况下仍能有效估计, 而此时子空间分解类算法若不作特别处理则将失效, 并且在只有一次快拍采样的条件下也能够估计出信号的来波方向, 但在信源间相关性增强时算法的性能恶化严重。

在传统的 DOA 估计算法被提出之后, 为了扩展其应用范围, 又有大量的新算法和改进算法涌现, 如改进最大似然估计方法、压缩感知算法改进的相干信源测向算法等^[7-10]。在空间谱估计中, 最大似然算法较为简单, 且在低信噪比、小快拍数及相干信源情况下都具有优异的性能。然而, 其实施的过程是复杂的, 它涉及一个多维非线性优化问题, 计算量相当大。随着智能计算的快速发展, 利用各种智能算法进行 DOA 估计, 也成为解决测向难题的一个有效途径^[11-17]。

8.1 经典 DOA 估计模型和算法

本节介绍了在高斯噪声环境下均匀线阵的测向模型, 给出了三种常见的高精度测向算法的数学模型, 并讨论了其使用量子蛙跳算法实现的过程, 给出了试验对比。

8.1.1 DOA 估计模型

对于 M 元均匀直线阵列, 假设阵元均为各向同性, 阵列远场有 N 个波长

为 λ 的窄带点源以平面波从 $\theta_i (i=1, 2, \dots, N)$ 入射, 阵元间距为 d , 阵列接收的第 l 次快拍数据可表示为

$$\mathbf{x}(l) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(l) + \mathbf{n}(l) \quad (8.1.1)$$

式中, $\mathbf{x}(l) = [x_1(l), x_2(l), \dots, x_M(l)]^T$ 为 $M \times 1$ 维阵列快拍数据矢量; $\mathbf{n}(l)$ 为 $M \times 1$ 维阵列噪声矢量, 阵列噪声假定为空间和时间均独立的高斯白噪声, 其均值为 0, 方差为 σ^2 ; $\mathbf{s}(l)$ 为 $N \times 1$ 维信号矢量; $\mathbf{A}(\boldsymbol{\theta})$ 为 $M \times N$ 维的导向矩阵, $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_N)]$, 其中导向矢量为 $\mathbf{a}(\theta_i) = [1, e^{-j2\pi d \sin(\theta_i)/\lambda}, \dots, e^{-j2\pi(M-1)d \sin(\theta_i)/\lambda}]^T (i=1, 2, \dots, N)$, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_N]$ 为信源的角度矢量。

接收到的阵列信号的协方差矩阵为 $\hat{\mathbf{R}} = \frac{1}{L} \sum_{l=1}^L \mathbf{x}(l)\mathbf{x}^H(l)$, 其中 L 是快拍数。

8.1.2 经典测向算法

1. 极大似然算法

根据确定性最大似然准则可以得到角度估计方程为

$$\hat{\theta}_1 = \arg \max_{\theta} \{\text{trace}(\mathbf{P}_{A(\theta)} \hat{\mathbf{R}})\} \quad (8.1.2)$$

其中, $\mathbf{P}_{A(\theta)} = \mathbf{A}(\boldsymbol{\theta})(\mathbf{A}^H(\boldsymbol{\theta})\mathbf{A}(\boldsymbol{\theta}))^{-1}\mathbf{A}^H(\boldsymbol{\theta})$ 为 \mathbf{A} 的映射矩阵。

2. 加权信号子空间拟合算法

对 $\hat{\mathbf{R}}$ 进行特征分解有 $\hat{\mathbf{R}} = \mathbf{U}_s \mathbf{F}_s \mathbf{U}_s^H + \mathbf{U}_n \mathbf{F}_n \mathbf{U}_n^H$, \mathbf{U}_s 代表由大特征值对应的特征矢量张成的信号子空间, \mathbf{U}_n 代表由小特征值对应的特征矢量张成的噪声子空间, \mathbf{F}_s 是 N 个大特征值组成的对角阵, \mathbf{F}_n 是小特征值组成的对角阵。得到的角度估计方程为

$$\hat{\theta}_2 = \arg \max_{\theta} \{\text{trace}(\mathbf{P}_{A(\theta)} \mathbf{U}_s \mathbf{W} \mathbf{U}_s^H)\} \quad (8.1.3)$$

其中, $\mathbf{W} = (\mathbf{F}_s - \mu \mathbf{I})^2 \mathbf{F}_s^{-1}$, μ 是 $M - N$ 个小特征值的平均值, \mathbf{I} 为单位阵。

3. 多维 MUSIC 算法

当 $\mathbf{W} = \mathbf{I}$ 时, 加权信号子空间拟合算法就成为多维 MUSIC 算法。其估计方程为

$$\hat{\theta}_3 = \arg \max_{\theta} \{\text{trace}(\mathbf{P}_{A(\theta)} \mathbf{U}_s \mathbf{U}_s^H)\} \quad (8.1.4)$$

8.1.3 基于量子蛙跳算法的测向方法

根据式 (8.1.2) ~ (8.1.4) 可以写出三种测向方法所对应的三个适应度函数, 分别为 $f_1(\theta) = \text{trace}(\mathbf{P}_{A(\theta)} \hat{\mathbf{R}})$ 、 $f_2(\theta) = \text{trace}(\mathbf{P}_{A(\theta)} \mathbf{U}_s \mathbf{W} \mathbf{U}_s^H)$ 和 $f_3(\theta) = \text{trace}(\mathbf{P}_{A(\theta)} \mathbf{U}_s \mathbf{U}_s^H)$ 。分别使用量子蛙跳算法求解上述适应度函数, 所对应的测向方法分别称作 QSFL-ML、QSFL-WSSF 和 QSFL-MUSIC 算法。

对于三种测向算法基于量子蛙跳算法的实现流程的统一框架可以归纳如下。

Step 01 参数初始化。首先初始化族群数量 m 、每个族群中的量子青蛙数量 n (种群规模 p 由 $p = m \times n$ 给定)、族群内的最大迭代次数 gen 及整个种群的最大迭代次数 Gen , 然后在量子定义域内随机产生青蛙的初始量子位置。

Step 02 将量子位置映射到相应位置, 计算每个青蛙的适应度值, 并根据适应度值将青蛙降序排列, 全局最优量子位置为 \mathbf{x}_g 。

Step 03 将全部群体分为 m 个族群, 其划分规则为: 第 1 只青蛙进入第 1 个族群, 第 2 只青蛙进入第 2 个族群, …… , 第 m 只青蛙进入第 m 个族群。然后, 第 $m+1$ 只青蛙又进入第 1 个族群, 依此类推, 直到全部青蛙划分完毕。

Step 04 设置 $i_m = 0$, 其中, i_m 代表族群数序号。

Step 05 令 $i_m = i_m + 1$, $i_{gen} = 0$, 其中, i_{gen} 代表族群内迭代次数的序号。族群内有最优适应度值和最差适应度值的青蛙量子位置分别为 \mathbf{x}_b 和 \mathbf{x}_w 。

Step 06 令 $i_{gen} = i_{gen} + 1$ 。

Step 07 在族群 i_m 内，按照量子蛙跳算法的更新策略更新最差适应度值的青蛙量子位置 \mathbf{x}_w ，更新方程为 $x_{wj}^{\text{new}} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}|$ ， $\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{bj}) + \text{randn} \cdot (x_{wj} - z_j)$ ；如果新的青蛙量子位置的适应度值没有改进，则在更新策略中用 \mathbf{x}_g 代替 \mathbf{x}_b ，即 $x_{wj}^{\text{new}} = |x_{wj} \cos \theta_{wj} - \sqrt{1 - (x_{wj})^2} \sin \theta_{wj}|$ ， $\theta_{wj} = \text{rand} \cdot (x_{wj} - x_{gj}) + \text{randn} \cdot (x_{wj} - z_j)$ ，使其更快地向全局最优解收敛；如果适应度值仍然没有改进，那么在量子域内随机产生新的青蛙量子位置 $\mathbf{x}_w^{\text{new}}$ 。这样便完成了族群 i_m 的一次循环，更新该族群内的 \mathbf{x}_b 、 \mathbf{x}_w 和 \mathbf{x}_g 。

Step 08 如果 $i_{\text{gen}} < \text{gen}$ ，则返回 **Step 06**；否则，进入 **Step 09**。

Step 09 如果 $i_m < m$ ，则返回 **Step 05**。

Step 10 判断是否满足终止条件。如果达到最大迭代次数，则输出最终 \mathbf{x}_g 及相应的最优解，终止算法；否则，按照适应度值降序排列青蛙，返回 **Step 03**。

8.1.4 试验仿真

量子蛙跳算法的参数设置如下：青蛙个体总数 $p=100$ ，族群数 $m=20$ ，每个族群青蛙数 $n=5$ ，族群内最大迭代次数 $\text{gen}=5$ ，整个种群的最大混洗迭代次数 $\text{Gen}=100$ 。使用 6 元均匀线阵的测向系统，阵元间距是半波长，考虑从两个方向（ $\theta_1 = 20^\circ$ 和 $\theta_2 = 10^\circ$ ）进入阵列的统计独立的信号，快拍数设定为 100。

图 8.1.1 给出了信噪比不同时三种算法的均方根误差性能。图 8.1.2 表明了信噪比不同时的估计成功概率（测量值和真实值之间的绝对偏差在 1° 之内设为估计成功）。可以看出，QSFL-ML 和 QSFL-WSSF 是优于 QSFL-MUSIC 的。虽然理论上 WSSF 优于 ML、ML 优于 MUSIC，但在使用智能优化算法进行一些实际仿真时，WSSF 和 ML 优于 MUSIC 是明显的，但是 WSSF 对于 ML 的优势在视觉上并不明显。

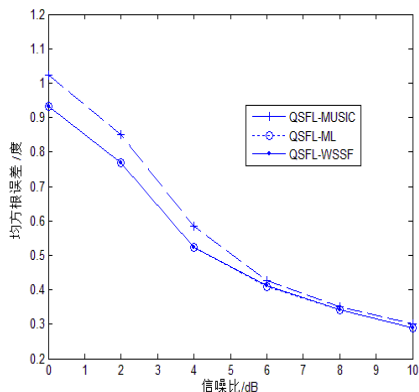


图 8.1.1 均方根误差与信噪比关系曲线

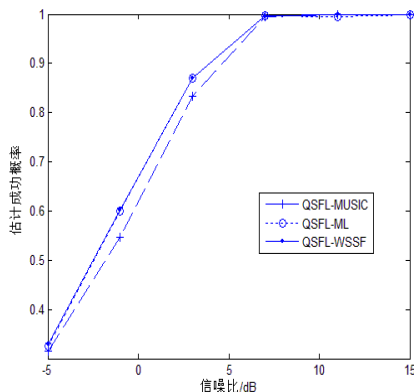


图 8.1.2 估计成功概率与信噪比关系曲线

8.2 基于高阶累积量和文化量子算法的测向方法^[18]

一直以来，基于高阶累积量的 DOA 估计技术大多研究的是独立信源，其扩展独立信源阵列孔径能力可以突破二阶统计量所要求的信号源数要小于阵元数的限制，使用较少的阵元数就可获得较高的检测性能^[19-21]。随着研究的深入，Gonen.E 提出了基于四阶累积量的相干信号 DOA 估计方法^[22]，该方法首先将相干信号的阵列输出模型等效为独立信号源，经过一个修正阵列的输出，并利用 TLS-ESPRIT 算法估计出修正阵列的方向矩阵，然后通过空间平滑技术对相干信号进行 DOA 估计。文献[23]提出了一种估计修正阵列方向矩阵的改进算法，该算法将一个高维的广义特征分解转化成低维的广义特征分解，然后通过模式激励和空间平滑技术相结合实现对相干信号的 DOA 估计。文献[22]和[23]虽然都能够对相干信号进行估计，但都使用了空间平滑技术，都是在损失有效扩展阵元数的代价下才获得了一定的解相干能力，由于受空间平滑技术的限制，只可对线阵或通过预处理转化成线阵的情况才有效。因此，找到不受天线阵的几何形状限制且不损失扩展天线阵列孔径的解相干高阶累积量测向算法有更实用的价值。

为了解决上述难题，在多个远场相干信号源的测向算法中引入合成导向矢

量,并结合四阶累积量的特点,精简扩展导向矢量和四阶累积量,提出了一种可实现相干和非相干信源有效测向的广义加权信号子空间拟合测向算法。为了有效求解所提测向算法,结合文化算法和量子计算,设计了文化量子算法进行精确求解。

8.2.1 基于高阶累积量的广义加权子空间拟合算法

假设在 M 元各向同性阵列(均匀线阵)的远场 $\theta_i(i=1,2,\dots,N)$ 处,有 N 个窄带点源以平面波入射,入射波长为 λ ,相邻天线间的距离 d ,则阵列接收的快拍数据为

$$\mathbf{x}(l) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(l) + \mathbf{n}(l) \quad (8.2.1)$$

其中, $\mathbf{x}(l)=[x_1(l),x_2(l),\dots,x_M(l)]^T$ 为 $M \times 1$ 维的阵列接收的快拍数据矢量; $\mathbf{s}(l)$ 为 $N \times 1$ 维的非高斯信号矢量; $\mathbf{A}(\boldsymbol{\theta})=[\mathbf{a}_1,\mathbf{a}_2,\dots,\mathbf{a}_N]$ 为 $M \times N$ 维的阵列流形矩阵, $\boldsymbol{\theta}=[\theta_1,\theta_2,\dots,\theta_N]$ 为信源的方向矢量; $\mathbf{n}(l)$ 为 $M \times 1$ 维的阵列噪声矢量,且阵列噪声假定为空间和时间均独立的高斯白噪声,其均值为 0,方差为 σ^2 。

若将空间信源分为 $Q(Q \leq N)$ 个相干源组,每个相干源组的信源个数分别为 d_1,d_2,\dots,d_Q , $d_1+d_2+\dots+d_Q=N$,且各相干源组的生成信源是相互独立的,则相干源组对应的信号生成源记为 $s_i(l)(i=1,2,\dots,Q)$ 。第 i 个相干源组中各相干源与生成源的相对幅度以矢量形式记为 $\mathbf{b}_i=[b_{i1},b_{i2},\dots,b_{id_i}]^T(i=1,2,\dots,Q)$,阵列接收快拍数据模型可以重新写作

$$\begin{aligned} \mathbf{x}(l) &= [\mathbf{A}_1\mathbf{b}_1,\mathbf{A}_2\mathbf{b}_2,\dots,\mathbf{A}_Q\mathbf{b}_Q]\mathbf{s}'(l) + \mathbf{n}(l) \\ &= [\mathbf{g}_1,\mathbf{g}_2,\dots,\mathbf{g}_Q]\mathbf{s}'(l) + \mathbf{n}(l) = \mathbf{G}\mathbf{s}'(l) + \mathbf{n}(l) \end{aligned} \quad (8.2.2)$$

其中, $\mathbf{g}_i=\mathbf{A}_i\mathbf{b}_i(i=1,2,\dots,Q)$ 、 $\mathbf{s}'(l)=[s_1(l),s_2(l),\dots,s_Q(l)]^T$ 为各相干源组的相互独立的生成源构成的信号矢量; \mathbf{A}_i 为第 i 个相干源组对应的导向矩阵; $\mathbf{g}_i(i=1,2,\dots,Q)$ 称为该子阵列的合成导向矢量,它是由相干源组内的导向矢量以相对幅度为权进行线性叠加构成的,矩阵 \mathbf{G} 被称为由合成导向矢量构成的合成导向矩阵。

由于合成信号与噪声不相关，合成信号间两两相关系数也不为 1，此时阵列接收数据的四阶累积量形式可表示为

$$\begin{aligned} C_{4x}(k_1, k_2, k_3, k_4) &= \text{cum}\{x_{k_1}, x_{k_2}, x_{k_3}^*, x_{k_4}^*\} \\ &= E\{x_{k_1} x_{k_2} x_{k_3}^* x_{k_4}^*\} - E\{x_{k_1} x_{k_3}^*\} E\{x_{k_2} x_{k_4}^*\} - \\ &\quad E\{x_{k_1} x_{k_4}^*\} E\{x_{k_2} x_{k_3}^*\} - E\{x_{k_1} x_{k_2}\} E\{x_{k_3}^* x_{k_4}^*\} \end{aligned} \quad (8.2.3)$$

可以简化为

$$C_{4x} = E\{(\mathbf{x} \otimes \mathbf{x})(\mathbf{x} \otimes \mathbf{x})^H\} - E\{(\mathbf{x} \otimes \mathbf{x})\} E\{(\mathbf{x} \otimes \mathbf{x})^H\} - E\{\mathbf{x} \mathbf{x}^H\} \otimes E\{\mathbf{x} \mathbf{x}^H\} \quad (8.2.4)$$

对于 M 元阵列而言，式中变量的取值范围为 $1 \leq k_1, k_2, k_3, k_4 \leq M$ ，因此随着 k_1, k_2, k_3, k_4 的变化，共有 M^4 个值。为了便于操作，需要将 M^4 个值放入如下 $M^2 \times M^2$ 的矩阵 \mathbf{R}_4 中：

$$\mathbf{R}_4((k_1 - 1)M + k_2, (k_3 - 1)M + k_4) = C_{4x}(k_1, k_2, k_3, k_4) \quad (8.2.5)$$

由于理想情况下噪声四阶累积量 \mathbf{R}_n 为 0，则可以得到

$$\mathbf{R}_4 = \mathbf{D}(\theta) \mathbf{C}_{s'} \mathbf{D}^H(\theta) + \mathbf{R}_n \quad (8.2.6)$$

其中

$$\mathbf{D}(\theta) = [\mathbf{g}_1 \otimes \mathbf{g}_1, \mathbf{g}_2 \otimes \mathbf{g}_2, \dots, \mathbf{g}_Q \otimes \mathbf{g}_Q] \quad (8.2.7)$$

$$\begin{aligned} \mathbf{C}_{s'} &= E\{(\mathbf{s}' \otimes \mathbf{s}')(\mathbf{s}' \otimes \mathbf{s}')^H\} - E\{(\mathbf{s}' \otimes \mathbf{s}')\} E\{(\mathbf{s}' \otimes \mathbf{s}')^H\} - \\ &\quad E\{\mathbf{s}' \mathbf{s}'^H\} \otimes E\{\mathbf{s}' \mathbf{s}'^H\} \end{aligned} \quad (8.2.8)$$

根据导向矩阵 \mathbf{D} 的结构可以知道导向矢量是有冗余的，其对应的四阶累积量中也相应地有大量的冗余元素，若直接用于测向则计算量较大。为简化计算，根据导向矩阵和四阶累积量的结构特点去除冗余进而缩小四阶累积量阵列维数。若四阶累积量扩展出的天线数为 M_a ，就应该有 M_a 行不同的导向矩阵矢量，则有 $M^2 - M_a$ 行导向矩阵矢量与其他行导向矩阵矢量重合。利用匹配搜索算法找到 $M^2 - M_a$ 个冗余导向矩阵的行矢量，删除冗余矢量元素形成新的 $M_a \times Q$ 维导向矩阵 \mathbf{E} ，记下该删除矢量的行标号，同时删除 $M^2 \times M^2$ 维四阶累积量 \mathbf{R}_4 中对应标号行和列上的元素，缩减的四阶累积量为 \mathbf{R}_E ，其维数缩减为 $M_a \times M_a$ 。

根据特征值分解可以得到四阶累积量 \mathbf{R}_E 的信号子空间 \mathbf{U}_s 和噪声子空间 \mathbf{U}_n ，则

$$\mathbf{R}_E = \mathbf{U}_s \mathbf{F}_s \mathbf{U}_s^H + \mathbf{U}_n \mathbf{F}_n \mathbf{U}_n^H \quad (8.2.9)$$

其中， \mathbf{U}_s 是由合成大特征值对应的合成特征矢量张成的信号子空间； \mathbf{U}_n 是由合成小特征值对应的合成特征矢量张成的噪声子空间； \mathbf{F}_s 是由 Q 个大特征值组成的对角阵； \mathbf{F}_n 是由小特征值组成的对角阵。

广义加权子空间拟合测向算法通过找到一个辅助矩阵 \mathbf{T} 将导向矩阵 $\mathbf{E}(\theta)$ 与 $\mathbf{U}_s \mathbf{W}^{1/2}$ 进行最小二乘意义上的拟合，即

$$\hat{\mathbf{b}}, \hat{\theta}, \hat{\mathbf{T}} = \arg \min_{\mathbf{b}, \theta, \mathbf{T}} \left\| \mathbf{U}_s \mathbf{W}^{1/2} - \mathbf{E} \mathbf{T} \right\|_F^2 \quad (8.2.10)$$

$\|\cdot\|_F^2$ 表示 Frobenius 范数， \mathbf{T} 为辅助参量， \mathbf{W} 为一个正定的加权矩阵。固定 \mathbf{E} 就可以求出 \mathbf{T} 的最小二乘解，将其带入式 (8.2.10)，就可得到信号参数 θ 的估计值，即下列最大化问题的解：

$$(\theta', \mathbf{b}') = \arg \max_{\theta, \mathbf{b}} \text{trace}[\mathbf{P}_E \mathbf{U}_s \mathbf{W} \mathbf{U}_s^H] \quad (8.2.11)$$

$$\mathbf{P}_E = \mathbf{E}(\mathbf{E}^H \mathbf{E})^{-1} \mathbf{E}^H \quad (8.2.12)$$

其中， \mathbf{P}_E 为导向矩阵 \mathbf{E} 的投影矩阵，最优权值为 $\mathbf{W} = (\mathbf{F}_s - \mu \mathbf{I})^2 \mathbf{F}_s^{-1}$ ， μ 为 $M_a - Q$ 个小特征值的平均， \mathbf{I} 为单位阵。称所提的算法为广义高阶加权子空间拟合算法，记作 GWSSF。

当信源相互独立时，广义高阶加权信号子空间拟合算法可简化为仅对角度估计，其估计方程为

$$\theta' = \arg \max_{\theta} \text{trace}[\mathbf{P}_E \mathbf{U}_s \mathbf{W} \mathbf{U}_s^H] \quad (8.2.13)$$

可以看出，独立信源的四阶累积量测向是广义高阶加权信号子空间拟合算法的一种特殊情况。在相干信源存在时，若对广义加权信号子空间拟合算法求解，则原有的轮换投影变换等求解加权子空间拟合算法不再有效。因此，结合文化算法和量子计算的理论设计新的优化算法来对方程进行有效求解是有重要意义的。

8.2.2 文化量子算法

利用量子旋转门独有的演进优势与文化算法相结合,将文化算法的影响函数与量子演进机理相结合提出文化量子算法(Cultural Quantum Algorithm, CQA),扩展了原文化算法进化规划种群模型的影响函数。文化量子算法的一个优点在于避免了现有演进算法的判决钳位操作,因为包括文化算法在内的大多数智能计算算法在进化过程中可能超出定义域,每次进化操作后需要判决置位,这需要大量计算,而量子旋转门在量子区间旋转,无须判决置位处理,更适合工程应用。因为要求解的测向问题为最大值优化问题,则所设计的算法关键部分如下。

1. 量子种群空间编码

种群空间采用量子种群模型。由于可行域为实数集,故量子个体可采用的量子编码结构为

$$\mathbf{Y}_j = \begin{bmatrix} \mathbf{y}_j \\ \boldsymbol{\beta}_j \end{bmatrix} = \begin{bmatrix} y_{j1}, y_{j2}, \dots, y_{jn} \\ \beta_{j1}, \beta_{j2}, \dots, \beta_{jn} \end{bmatrix} \quad (8.2.14)$$

其中, $\mathbf{Y}_{ji} = [y_{ji}, \beta_{ji}]^T$ 为第 j 个量子个体的第 i 维量子态表示方式, y_j 代表文化量子个体, $0 \leq y_{ji} \leq 1$, $0 \leq \beta_{ji} \leq 1$, 且满足 $y_{ji}^2 + \beta_{ji}^2 = 1$ 。

2. 量子信仰空间结构

量子信仰空间的结构采用 $\langle \mathbf{H}, \mathbf{Q} \rangle$ 结构, 其中 $\mathbf{H} = \{\mathbf{h}'\}$ 为量子形势知识, \mathbf{h}' 表示到第 t 代为止所找到的最优文化量子个体; $\mathbf{Q} = \langle \mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_n \rangle$ 为量子规范知识, 表示量子变量取值区间的信息, n 为待求量子变量维数, \mathbf{Q}_i 为 $\langle \mathbf{I}_i, L_i, U_i \rangle$, $\mathbf{I}_i = [l_i, u_i]$, 下限 l_i 和上限 u_i 根据问题所给定的量子变量定义域来初始化; L_i 表示第 i 维量子规范知识的下限 l_i 所对应的评价值, U_i 表示第 i 维量子规范知识的上限 u_i 所对应的评价值, 均初始化为 $-\infty$ 。

3. 接受函数

接受函数用于选择能够直接影响当前量子信仰空间知识经验的量子个体。

即从当前量子种群空间中以 α 比例选择最优量子个体，即

$$\text{Accept}() = \lceil m \cdot \alpha \rceil \quad (8.2.15)$$

其中， $\lceil \cdot \rceil$ 为取整函数。

4. 量子信仰空间的更新

量子信仰空间由更新函数进行调整，更新函数主要包括量子形势知识更新函数和量子规范知识更新函数。第 $t+1$ 代量子形势知识 \mathbf{h}^{t+1} 由上一代进化得到的最优文化量子个体 $\mathbf{y}_{\text{best}}^t$ 和量子形势知识依照量子形势知识更新函数来更新。

$$\mathbf{h}^{t+1} = \begin{cases} \mathbf{y}_{\text{best}}^t, & f(\bar{\mathbf{y}}_{\text{best}}^t) > f(\bar{\mathbf{h}}^t) \\ \mathbf{h}^t, & \text{其他} \end{cases} \quad (8.2.16)$$

其中， $\mathbf{y}_{\text{best}}^t$ 表示第 t 代最优文化量子个体； $\bar{\mathbf{y}}_{\text{best}}^t$ 为最优文化量子个体的映射态； $\bar{\mathbf{h}}^t$ 为第 t 代量子形势知识的映射态。

设第 j 个量子个体影响量子规范知识 \mathbf{Q} 的下限，第 k 个量子个体影响量子规范知识 \mathbf{Q} 的上限，通过下面的量子规范知识更新函数对量子规范知识 \mathbf{Q} 进行更新。

$$l_i^{t+1} = \begin{cases} y_{ji}^t, & l_i^t \text{ 或 } f(\bar{\mathbf{y}}_j^t) > L_i^t \\ l_i^t, & \text{其他} \end{cases} \quad (8.2.17)$$

$$L_i^{t+1} = \begin{cases} f(\bar{\mathbf{y}}_j^t), & l_i^t \text{ 或 } f(\bar{\mathbf{y}}_j^t) > L_i^t \\ L_i^t, & \text{其他} \end{cases} \quad (8.2.18)$$

$$u_i^{t+1} = \begin{cases} y_{ki}^t, & u_i^t \text{ 或 } f(\bar{\mathbf{y}}_k^t) > U_i^t \\ u_i^t, & \text{其他} \end{cases} \quad (8.2.19)$$

$$U_i^{t+1} = \begin{cases} f(\bar{\mathbf{y}}_k^t), & u_i^t \text{ 或 } f(\bar{\mathbf{y}}_k^t) > U_i^t \\ U_i^t, & \text{其他} \end{cases} \quad (8.2.20)$$

其中， L_i^t 和 U_i^t 分别表示第 t 代第 i 维量子规范知识的下限 l_i^t 和上限 u_i^t 所对应的评价价值； $\bar{\mathbf{y}}_j$ 和 $\bar{\mathbf{y}}_k$ 分别为文化量子个体 \mathbf{y}_j 和 \mathbf{y}_k 映射到定义域区间的潜在解向量。

5. 量子旋转门实现的影响函数

由于不同文化算法的区别主要在于所采取的影响函数不同^[24]，文化量子算法根据所特有的量子形势知识和量子规范知识，使用量子旋转门完成文化算法的演进，设计了两种影响函数完成父代文化量子个体的进化。量子信仰空间能够通过改变量子旋转角的步长和前进方向来影响量子种群空间的量子变量变异的质量和效果。

使用量子旋转门的影响函数可以定义为

$$\mathbf{Y}_{ji}^{t+1} = \text{abs}(\mathbf{U}_{ji}^t \mathbf{Y}_{ji}^t) \quad (8.2.21)$$

其中， $\mathbf{Y}_{ji}^t = [y_{ji}^t, \text{sqrt}(1 - y_{ji}^t y_{ji}^t)]^T$ ，量子旋转门为

$$\mathbf{U}_{ji}^t = \begin{bmatrix} \cos \phi_{ji}^t & -\sin \phi_{ji}^t \\ \sin \phi_{ji}^t & \cos \phi_{ji}^t \end{bmatrix} \quad (8.2.22)$$

使用量子形势知识调整量子变量变化步长和前进方向，其量子旋转角为

$$\phi_{ji}^t = \begin{cases} -\frac{\pi}{2} |N(0,1) \text{size}(\mathbf{I}_i^t)|, & y_{ji}^t < h_i^t \\ \frac{\pi}{2} |N(0,1) \text{size}(\mathbf{I}_i^t)|, & y_{ji}^t > h_i^t \\ \frac{\pi}{2} \eta N(0,1) \text{size}(\mathbf{I}_i^t), & \text{其他} \end{cases} \quad (8.2.23)$$

其中， $\text{size}(\mathbf{I}_i^t)$ 为量子信仰空间中第 i 维量子变量在第 t 代可调整量子区间的长度； η 为缩放因子； $N(0,1)$ 代表均值为 0、方差为 1 的高斯随机数。

使用量子规范知识调整量子变量变化步长及前进方向的影响函数，其量子旋转角定义如下：

$$\phi_{ji}^t = \begin{cases} -\frac{\pi}{2} |N(0,1) \text{size}(\mathbf{I}_i^t)|, & y_{ji}^t < u_i^t \\ \frac{\pi}{2} |N(0,1) \text{size}(\mathbf{I}_i^t)|, & y_{ji}^t > u_i^t \\ \frac{\pi}{2} \eta N(0,1) \text{size}(\mathbf{I}_i^t), & \text{其他} \end{cases} \quad (8.2.24)$$

该影响函数可以使父量子个体处在较好的量子区间时仅做微小变动，在其他情况下则使父量子个体的变异尽可能朝向信仰空间中量子规范知识所限定的区间移动。

8.2.3 基于文化量子算法的广义高阶加权信号子空间拟合测向

在要解决的测向问题中， θ 和 b 为待求矢量，则文化量子算法的适应度函数为

$$f(\theta, b) = \text{trace}[P_E U_s W U_s^H] \quad (8.2.25)$$

确定了适应度函数，则测向问题就转化为对文化量子算法中最优量子个体映射态的求解。利用文化量子算法实现广义高阶加权子空间拟合 (CQ-GWSSF) 测向的过程描述如下。

Step 01 量子种群空间初始化，生成 m 个 n 维量子向量，对于独立信源测向，所需求解向量由角度组成；对于相干信源，为来波方向角和相对幅度的联合矢量，这样就产生了规模为 m 的初始量子种群空间。

Step 02 把文化量子个体映射到待求解的定义域，计算映射态的适应度，对量子种群空间中的个体进行评价。

Step 03 根据给定的量子取值范围和初始量子种群空间中的候选解，按照量子信仰空间的结构，生成初始量子信仰空间。

Step 04 每个文化量子个体依概率选择不同的量子旋转角，根据选择的影响函数对量子种群空间中的每个父文化量子个体使用量子旋转门进行变异，生成 m 个相应的子文化量子个体。

Step 05 把子文化量子个体映射到待求解的定义域，进行适应度评价，对由子文化量子个体和父文化量子个体共同组成的规模为 $2m$ 的量子种群空间，选择前 m 个优秀文化量子个体作为下一代的父文化量子个体。

Step 06 按照设定的接受函数选择最优文化量子个体，并按式 (8.2.16) ~ 式 (8.2.20) 来更新量子信仰空间。

Step 07 判断是否满足迭代终止条件。若满足，则输出当代最优文化量子个体的映射态；如果不满足，则返回 **Step 04**，继续循环。

8.2.4 试验仿真

仿真中用于比较的算法有：CQ-WSSF 代表文化量子算法实现的加权子空间拟合测向方法；MUSIC-LIKE 是使用四阶累积量的类 MUSIC 算法；CA-ML 代表文化算法实现的极大似然算法^[25]；CPSO-GML 代表基于混沌自适应变异粒子群算法的广义极大似然测向算法^[26]。其中，MUSIC、CA-ML 和 CQ-WSSF 算法不能实现阵列扩展；而 MUSIC-LIKE 算法只能实现独立信源的阵列扩展，不能测相干信源；CPSO-GML 算法能测多个相干信源，却不能实现对独立信源的阵列扩展。在文化量子算法中接受函数的选取比例为 $\alpha = 0.2$ ，缩放因子取 $\eta = 0.06$ 。为验证基于文化量子算法的广义高阶信号子空间拟合 (CQ-GWSSF) 测向算法的有效性和优越性，进行如下仿真试验。

试验一：独立信源测向

考虑阵元数为 3 的等距均匀线阵，阵元间距为 $1/2$ 波长，快拍数取 1024，两个独立窄带信号从 4.45° 和 16.15° 方向入射到该等距线阵。每个信噪比下独立进行了 400 次 Monte Carlo 仿真试验。文化算法和文化量子算法的群体规模都设为 100，终止迭代次数为 100。从图 8.2.1 和图 8.2.2 中可以看出，CQ-GWSSF 的均方根误差和估计成功概率（测量值和真实值之间的绝对偏差在 1° 之内设为估计成功）都优于 MUSIC、CQ-WSSF、CA-ML 和 MUSIC-LIKE。在对独立信源测向时，广义极大似然算法 (GML) 就退化为极大似然算法，CPSO-GML 的性能近似于 CA-ML，故在此处不仿真 CPSO-GML 算法。

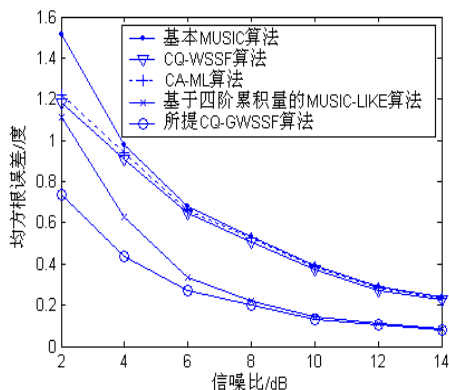


图 8.2.1 均方根误差与信噪比关系曲线

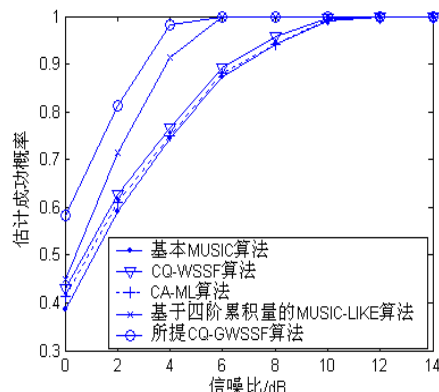


图 8.2.2 估计成功概率与信噪比关系曲线

试验二：相干信号测向

考虑阵元数为 3 和 5 的均匀线阵，快拍数取 1024，两个相干信号分别从 0.45° 和 14.55° 入射到该阵列，其性能仿真曲线如图 8.2.3 和图 8.2.4 所示。文化算法、CPSO 算法和文化量子算法的种群规模设为 100，终止迭代次数设为 150。从仿真图可以看出，无论是用 3 根天线还是 5 根天线，CQ-GWSSF 的性能都是最优秀的，是一种高性能的去相干算法。

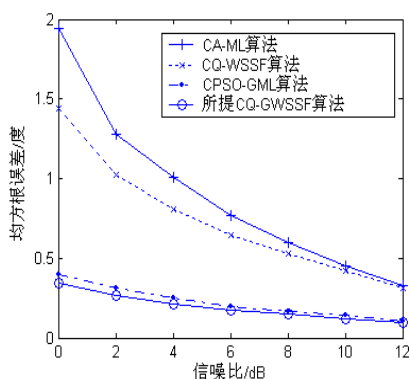


图 8.2.3 3 根天线测向性能比较

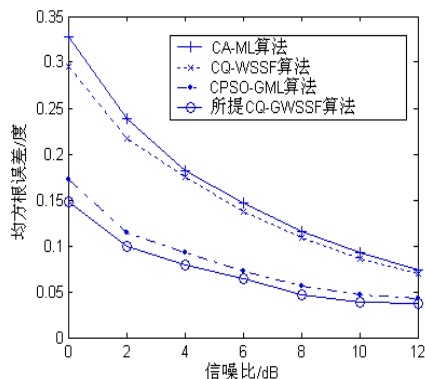


图 8.2.4 5 根天线测向性能比较

试验三：阵列扩展性能仿真

考虑阵元数为 3 的等距均匀线阵，快拍数取 1024，4 个远场信号分别从方位角 $\{50^\circ, 20^\circ, -10^\circ, -40^\circ\}$ 入射到该阵列，信噪比以 2 dB 为间隔从 0 dB 变化到

12dB, 每个信噪比下独立进行了 400 次 Monte Carlo 试验, 此处假设相干信源的相对幅度已知。图 8.2.5 和图 8.2.6 分别给出了测量 4 个独立信源和 4 个相干信源 (两个相干信源组, 50° 和 20° 方向信源完全相干, -10° 和 -40° 方向信源完全相干) 的仿真图。在测量独立信源时, CQ-GWSSF 的性能优于 MUSIC-LIKE; 在测量相干信源时, 在相同的种群规模和最大迭代次数情况下, CQ-GWSSF 的性能优于 CPSO-GML; MUSIC-LIKE 算法仅能对独立信源实现阵列扩展, 不能测量信源数大于阵元数的相干信源; CPSO-GML 仅能对相干信源实现阵列扩展, 对独立信源不能实现阵列扩展。所提出的 CQ-GWSSF 算法能实现对独立和相干信源测向时的阵列扩展, 而且性能较为优秀。

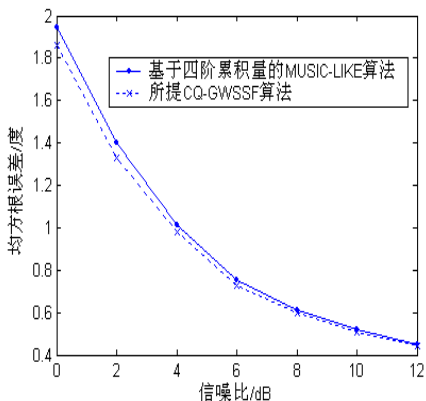


图 8.2.5 独立信源阵列扩展性能比较

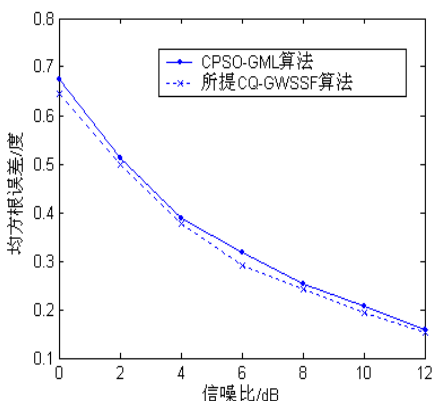


图 8.2.6 相干信源阵列扩展性能比较

在已知的各种测向算法中, MUSIC、CA-ML 和加权子空间拟合测向算法不能实现阵列扩展; 而 MUSIC-LIKE 算法只能实现独立信源阵列扩展, 不能测量相干信源, 更没有阵列扩展能力; CPSO-GML 算法能测量多余天线数的相干信源, 却不能实现对独立信源的阵列扩展; 在测量不超过阵元数的多个独立信源时, CPSO-GML 的性能近似等同于 CA-ML。总之, 现有测向算法都存在一些应用局限, 所提出的 CQ-GWSSF 测向算法能克服现有测向算法的应用局限, 并且具有更优秀的性能。所提出的文化量子算法的文化进化机制不是目前广泛使用的简单高斯变异的进化规则, 而是通过量子旋转门和量子信仰空间更好地为文化量子个体提供的演进信息, 保证在求解优化问题时快速收敛到最优。文

化量子算法避免了现有文化算法的判决钳位操作，因为包括文化算法在内的现有大多数用于实数优化的智能计算算法（如粒子群和遗传算法等）在进化过程中变量值可能超出定义域，每次进化操作后需要判决置位，浪费了计算时间；而量子旋转门在量子区间旋转，无须判决置位操作，因此更适合编程和工程应用。

8.3 基于量子文化蛙跳算法的非圆信号 DOA 估计^[27]

在 20 世纪 90 年代末，一些学者发现一些通信信号具有非圆特性，利用非圆特性可以提高经典测向算法的测向性能，因此提出了非圆信号测向的类 MUSIC 算法和类 ESPRIT 算法^[28-30]。由于利用了信号的非圆特性，可以看作增加了天线阵的有效阵元数，使得经典测向算法在非圆信号测向的时候分辨能力和测向精度都得到了提高。现有的非圆信号测向方法及其处理过程大多与经典测向算法类似，没有充分利用非圆信号的非圆信息，而且难以实现对相干信源的测向。因此，充分利用非圆信息，使用量子文化蛙跳算法设计出有合理计算量的相干和非相干非圆信号测向方法有重要的实用价值。

8.3.1 非圆极大似然算法

假设有一个由 M 个共向同性全向天线构成的均匀线阵， N 个波长为 λ 的窄带源从 $\theta_k (k=1,2,\dots,N)$ 方向入射到线阵。相邻阵元之间的距离为 d 。对于第 t 次快拍， M 个阵元接收数据为

$$\mathbf{y}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t) \quad (8.3.1)$$

其中， $\mathbf{y}(t)=[x_1(t), x_2(t), \dots, x_M(t)]^T$ 为 $M \times 1$ 维的阵列接收的快拍数据矢量； $\mathbf{s}(t)$ 为 $N \times 1$ 维的信号矢量； $\mathbf{A}(\boldsymbol{\theta})=[\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_N)]$ 为 $M \times N$ 维的阵列流形矩阵， $\boldsymbol{\theta}=[\theta_1, \theta_2, \dots, \theta_N]$ 为信源的方向矢量， $\mathbf{a}(\theta_i)=[1, e^{-j2\pi d \sin(\theta_i)/\lambda}, \dots, e^{-j2\pi(M-1)d \sin(\theta_i)/\lambda}]^T$ ($i=1,2,\dots,N$) 为第 i 个导向矢量； $\mathbf{n}(t)=[n_1(t), n_2(t), \dots, n_M(t)]^T$

为 $M \times 1$ 维的复高斯噪声矢量，且噪声方差为 σ^2 。

根据非圆率 $\rho = 1$ 时的非圆信号特性，接收到的数据可表示为

$$Y(t) = \begin{bmatrix} y(t) \\ y^*(t) \end{bmatrix} \quad (8.3.2)$$

则阵列接收的非圆信号的协方差矩阵为

$$C = E[Y(t)Y^H(t)] \quad (8.3.3)$$

在均匀线性阵列中非圆信号的导向矩阵可以表示为

$$B(\theta, \varphi) = \begin{bmatrix} A \\ A^* E^* \end{bmatrix} = [b(\theta_1, \phi_1), b(\theta_2, \phi_2), \dots, b(\theta_N, \phi_N)] \quad (8.3.4)$$

其中， $\varphi_i (i=1, 2, \dots, N)$ 是非圆相位。第 i 个方向为 $\theta_i (i=1, 2, \dots, N)$ 的导向矢量由下式给出

$$b(\theta_i, \phi_i) = \begin{bmatrix} a(\theta_i) \\ a^*(\theta_i) e^{-j\phi_i} \end{bmatrix} \quad (8.3.5)$$

非圆最大似然算法 (NC-ML) 可以写成方向向量的联合优化形式，即

$$\hat{\theta}, \hat{\varphi} = \arg \max_{\theta, \varphi} \text{trace}[P_{B(\theta, \varphi)} C] \quad (8.3.6)$$

其中， $P_{B(\theta, \varphi)} = B(\theta, \varphi)(B^H(\theta, \varphi)B(\theta, \varphi))^{-1}B^H(\theta, \varphi)$ 是 $B(\theta, \varphi)$ 的投影矩阵。

8.3.2 基于量子文化蛙跳算法的极大似然测向

QCSFL 算法的量子位置的初始量子位是在 $[0, 1]$ 间随机生成的。对于 NML 问题，每个量子青蛙的量子位置和位置是不同的。所以，NML 的目标函数可看作 QCSFL 的适应度函数，量子位置的适应度可以根据位置 (θ, φ) 的适应度计算得到，即

$$f(\theta, \varphi) = \text{trace}[P_{B(\theta, \varphi)} C] \quad (8.3.7)$$

因此，测向问题可以转化成利用 QCSFL 搜寻最优位置的问题。量子文化蛙跳测向方法的描述如下。

Step 01 参数初始化。首先初始化族群数量 m 、每个族群中的量子青蛙数量 n （种群规模 p 由 $p = m \times n$ 给定）、混洗前每个族群的最大迭代次数 q 及整个种群的最大迭代次数，然后在定义域内随机产生青蛙的初始量子位置，并利用量子文化的概念初始化量子信仰空间。

Step 02 把量子位置映射到位置，计算每只量子青蛙的适应度，然后降序排列。

Step 03 根据接受函数，选择一些优秀的量子青蛙，更新信仰空间的量子规范知识。通过混洗规则将整个种群分成 m 个族群，适应度最优的量子青蛙的量子位置定义为 \mathbf{x}_g 。

Step 04 令 $i_m = 0$ 。

Step 05 令 $i_m = i_m + 1$ ，其中 i_m 表示族群的标号。在该族群中分别记录下最优的量子位置 \mathbf{x}_b 和最差的量子位置 \mathbf{x}_w ，在该族群中计算出量子位置的均值为 \mathbf{z} ，令 $i_q = 0$ 。

Step 06 令 $i_q = i_q + 1$ ，其中 i_q 表示族群内迭代次数的序号。

Step 07 创建子族群。用之前描述的新的量子搜索策略调整最差量子青蛙的量子位置，计算 $\mathbf{x}_w^{\text{new}}$ 的适应度，在该族群中分别记录下最优的量子位置 \mathbf{x}_b 和最差的量子位置 \mathbf{x}_w 。若 $\mathbf{x}_w^{\text{new}}$ 的适应度优于 \mathbf{x}_g 的适应度，则 $\mathbf{x}_g = \mathbf{x}_w^{\text{new}}$ 。

Step 08 若 $i_q < q$ ，则返回 **Step 06**；否则，进入下一步。

Step 09 若 $i_m < m$ ，则返回 **Step 05**；否则，进入下一步。

Step 10 判断是否满足终止条件。如果达到最大迭代次数，则输出最终位置 \mathbf{x}_g 及相应的适应度值，终止算法；否则，将所有量子青蛙根据适应度降序排列，返回 **Step 03**。

8.3.3 试验仿真

为了评估提出的 QCSFL-NML 和之前一些经典算法的相对性能,进行了一系列仿真实验,例如 NC-MUSIC^[30]、CA-ML^[25]、CSFL-WSSF^[31]等。QCSFL-NML、NC-MUSIC、CSFL-WSSF 和 CA-ML 等算法使用五元均匀线阵($d = \lambda / 2$),所有的测向算法都在相同的环境下进行仿真。4 种智能算法的群体大小都设置为 100,最大迭代次数设置为 100。CSFL 和 QCSFL 算法的参数设置如下:族群的数量设置为 $m=20$;每个族群中量子青蛙的个数设置为 $n=5$;在混排后每个族群内的进化次数设置为 $q=5$;最大混排迭代次数设置为 $Gen=100$;由于问题维数不算高,故可取 $p_1 = p_2 = 1$, $p_3 = 1$ 。200 次的试验结果用于测试提出的智能算法的性能。

在非圆信号的情况下, QCSFL-NML 的均方根误差优于 ML, 因为 QCSFL-NML 算法扩展了阵列孔径。考虑从两个方向($\theta_1 = 21^\circ$ 和 $\theta_2 = 26.5^\circ$)进入阵列的统计独立的 BPSK 信号,快拍数设定为 300。图 8.3.1 表明了当信噪比(SNR)以 5dB 为间隔从 0dB 变化到 15dB 时 4 种算法的均方根误差;图 8.3.2 表明了当 SNR 以 5dB 为间隔从 -15dB 变化到 20dB 时 4 种算法的成功概率。图 8.3.1 和图 8.3.2 均表明,对于独立非圆信号而言, QCSFL-NML 算法的性能优于 CA-ML、CSFL-WSSF 和 NC-MUSIC 算法。

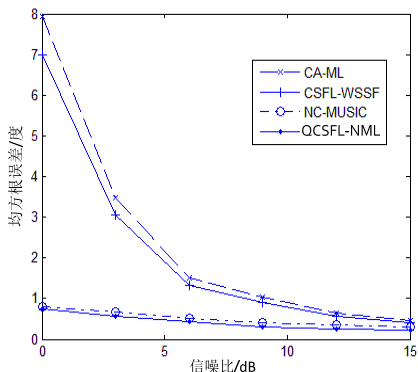


图 8.3.1 独立非圆信号 SNR 与 RMSE 曲线

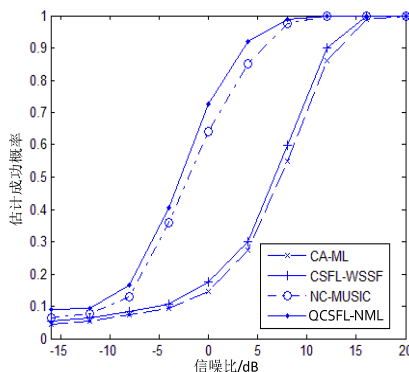


图 8.3.2 独立非圆信号 SNR 与成功概率的对比曲线

在两个相干非圆信号情况下 ($\theta_1 = 15^\circ$ 和 $\theta_2 = 29^\circ$) 进行本次仿真, 假定快拍数为 300。图 8.3.3 和图 8.3.4 表明, 对于相干非圆信号而言, QCSFL-NML 算法的均方根误差和成功概率均优于 CA-ML、CSFL-WSSF 和 NC-MUSIC 算法。在相干和非相干信号情况下, QCSFL-NML 算法的均方根误差和成功概率优于 CA-ML 和 CSFL-WSSF 算法, 这是因为 QCSFL-NML 算法利用了非圆信息和 QCSFL 算法, 因此它可以扩展阵列孔径并提高 ML 算法的性能。

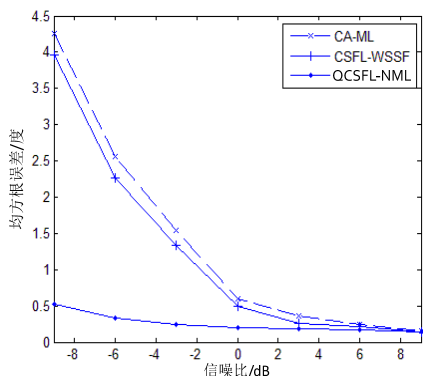


图 8.3.3 相干非圆信号 SNR 与 RMSE 曲线

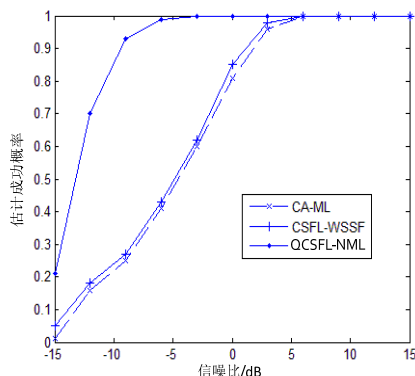


图 8.3.4 相干非圆信号 SNR 与成功概率的对比曲线

8.4 小结

本章主要介绍了高斯噪声背景下的测向模型和算法, 主要讨论了基于文化量子算法的广义高阶加权子空间拟合测向算法和基于量子文化蛙跳算法的非圆信号测向方法。基于文化量子算法的广义高阶加权子空间拟合测向算法可实现相干信源测向并可扩展阵列, 突破了现有的基于四阶累积量测向算法的局限, 且有非常优秀的测向性能, 优于现有的一些经典算法, 在复杂的测向环境中具有重要的价值。所提的文化量子算法是一种解决实数优化问题的新算法, 且在进化过程中无须判决钳位操作, 适合工程应用, 可进一步扩展到其他工程优化问题中。基于量子文化蛙跳算法的极大似然非圆信号测向算法, 能够充分利用非圆信号信息, 且所提方法可以测相干非圆信号的方向, 有更优秀的测向性能。

参考文献

- [1] 王永良. 空间谱估计理论与算法. 北京: 清华大学出版社, 2004.11.
- [2] Schmidt R. O. Multiple emitter location and signal parameter estimation. IEEE Trans, 1986, AP-34(3): 276-280.
- [3] Roy R, Kailath T. ESPRIT-estimation of signal parameters via rotational invariance techniques. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1989, 37(7): 984-995.
- [4] Stoica P, Nehorai A. MUSIC, maximum likelihood, and Cramer-Rao bound. In Proc. ICASSP, 1988: 2296-2299.
- [5] Ottersten B, Viberg M, Stoica P, Nehorai A. Exact and large sample ML techniques for parameter estimation and detection in array processing. IN Haykin, Litva, and shepherd, editors, Radar array processing, Springer-Verlag, Berlin, 1993: 99-151.
- [6] Viberg M, Ottersten B. Sensor array processing based on subspace fitting. IEEE Transactions on Signal Processing, 1991,39(5):1110-1121.
- [7] 刁鸣, 高璐, 高洪元, 等. 基于非均匀线阵的压缩感知波达方向估计. 计算机工程, 2015, 41 (10): 83-87.
- [8] 刁鸣, 李晓刚, 高洪元. 基于四阶累积量的最大似然测向方法研究. 电子信息对抗技术, 2007, 22 (5): 4-6.
- [9] 何昭然, 高洪元. 文化鱼群算法的广义 MUSIC 测向技术. 解放军理工大学学报, 2014, 15 (1): 7-11.
- [10] 安春莲, 赵振宇, 刁鸣, 等. 独立信号与相干信号并存的任意阵列测向新方法. 哈尔滨工程大学学报, 2013: 517-523.

- [11] He Z Y, Zhao Z Q, Yang K, Ouyang J. Power and DOA estimation for array processing using particle swarm optimization. Proc. International Conf. on Computational Problem-Solving, 2011: 405-408.
- [12] Sit Y L, Agatonovic M, Zwick T. Neural network based direction of arrival estimation for a MIMO OFDM radar. Proc. 9th European Radar Conf., 2012: 298-301.
- [13] Shi W, Huang J, Hou Y S. Fast DOA estimation algorithm for MIMO sonar based on ant colony optimization. Journal of Systems Engineering and Electronics, 2012, 23: 173-178.
- [14] 庞伟正, 高洪元, 王艳丽. 基于粒子群优化算法的相干信源波达方向估计. 哈尔滨工程大学学报, 2006, 27 (3): 453-456.
- [15] Gao Hongyuan, Han Xiaodong. Direction finding of signal subspace fitting based on cultural bee colony algorithm. The IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications, 2010: 966-970.
- [16] 刁鸣, 高洪元, 王艳温. 二维相干信源测向新方法研究. 系统工程与电子技术, 2006, 28 (9): 1289-1293.
- [17] Gao Hongyuan, Yu Xuemei, Cao Jinlong. Direction finding of maximum likelihood algorithm using quantum bee colony for noncircular signals. IEEE 10th International Conference on Signal Processing, 2010: 365-368.
- [18] 高洪元, 刁鸣. 文化量子算法实现的广义加权子空间拟合测向. 电波科学学报, 2010, 25 (4): 798-804.
- [19] Porat B, Friedlander B. Direction finding algorithm based on high order statistics. IEEE Transaction signal processing, 1991, 39(9): 2016-2024.
- [20] 林刚, 许家栋, 樊寄松. 对四阶累积量 MUSIC 算法的分析与应用. 电波科学学报, 2006, 21 (3): 357-360.

- [21] 唐建红, 司锡才, 彭巧乐. 快速四阶累积量旋转不变子空间算法. 西安交通大学学报, 2009, 43 (6): 88-92.
- [22] Gonen E., Mendel J. M., Dogan M. C. . Application of cumulants to array processing Part IV:direction finding in coherent signals case. IEEE Trans. on SP, 1997, 45(9): 2252-2264.
- [23] 王鼎, 吴瑛. 一种基于四阶累积量的相干信号测向算法. 系统工程与电子技术, 2006, 28 (5): 665-669.
- [24] 高洪元, 刁鸣. 数字滤波器设计的文化量子算法. 计算机应用, 2010, 30 (5): 1410-1414.
- [25] 刁鸣, 李晓刚, 王冰. 文化算法的最大似然测向方法研究. 哈尔滨工程大学学报, 2008, 29 (5): 509-513.
- [26] 张陆游, 张永顺, 杨云. 基于混沌自适应变异粒子群优化的解相干算法. 电子与信息学报, 2009, 31 (8): 1825-1829.
- [27] Gao Hongyuan, Xu Congqiang. Cultural quantum-inspired shuffled frog leaping algorithm for direction finding of non-circular signals. International Journal of Computing Science and Mathematics, 2013, 4(4): 321-331.
- [28] Charge P, Wang Y, Saillard J. A noncircular sources direction finding method using polynomial rooting. Signal Processing, 2001, 81(8): 1765-1770 .
- [29] Haardt M, Romer F. Enhancements of unitary ESPRIT for non-circular sources. In Proceedings of the ICASSP, 2004, 2: 101-104.
- [30] Abeida H, Delmas J P. MUSIC-like estimation of direction of arrival for noncircular sources. IEEE Trans. Signal Process, 2006, 56(7): 2678-2690.
- [31] Gao Hongyuan, Diao ming. Direction finding of weighted signal subspace fitting based on cultural shuffled frog leaping. Journal of Computational Information Systems, 2010, 6(3): 847-854.

第 9 章

冲击噪声环境下的量子智能计算 DOA 估计

经典的 DOA 估计理论大都是在假定阵元噪声为空时独立的高斯白噪声的前提下提出的。然而，在实际的应用环境中，许多情况是具有冲激特性的非高斯噪声，如水下噪声、低频大气噪声和若干人造噪声等，都可用不同特征指数 α 的 SaS 过程来描述^[1]。SaS 稳定分布是高斯分布的一种推广，Pearson 分布 ($\alpha=0.5$)、高斯分布 ($\alpha=2$) 和柯西分布 ($\alpha=1$) 只是其特例。特征指数 α 的取值范围为 $0 < \alpha \leq 2$ ，它决定了分布的形状，其值越小，概率密度函数的拖尾就越大，其冲击性就越强。然而，非高斯 SaS 噪声不具有二阶及以上的高阶矩，所以应用于高斯模型的二阶及高阶矩方法通常不能应用于非高斯 SaS 分布，特征指数 α 为 $[1,2]$ 间的弱冲击噪声背景下的测向需使用共变矩、分数低阶矩或分数低阶协方差矩阵^[2-4]。而且，强冲击噪声是指特征指数 α 为 $[0,1]$ 间的最恶劣的冲击噪声环境，在此区间现有的一些分数低阶矩类算法失效。以冲击噪声背景为测向研究对象，设计更广义的测向方法，提高冲击噪声环境下测向算法的鲁棒性，会极大地拓展现有测向方法的应用范围。

本章首先给出了冲击噪声环境下可有效抗噪声干扰的三种低阶矩，并对其进行了简单的比较；然后设计了量子文化细菌觅食算法，并在冲击噪声环境下

给出了一种基于量子文化细菌觅食算法的非圆测向方法；最后，针对冲击噪声环境下的动态跟踪问题，设计了基于量子粒子群算法的动态跟踪方法，仿真结果表明跟踪效果良好。

9.1 冲击噪声环境下的测向模型

由于非高斯 S α S 噪声不具有二阶及以上矩，因而在冲击噪声环境下需要计算接收信号的低阶矩阵，再结合传统的测向方法，完成冲击噪声环境下的鲁棒测向^[3-9]。最常见的低阶矩阵介绍如下，并和 MUSIC 算法结合验证其优劣。

9.1.1 三种低阶矩

对于 M 元均匀线阵，假设均为各向同性的阵元，阵元间距为 d ，在阵列远场处有 N 个窄带点源以平面波入射，入射波长为 λ ，则阵列接收的快拍数据为

$$\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t) \quad (9.1.1)$$

其中， $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_M(t)]^T$ 为 $M \times 1$ 维的阵列快拍数据矢量； $\mathbf{s}(t)$ 为 $N \times 1$ 维的信号矢量； $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_N)]$ 为阵列的流形矩阵， $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_N]$ 为信源的方位矢量， $\mathbf{a}(\theta_i) = [1, e^{-j2\pi d \sin(\theta_i)/\lambda}, \dots, e^{-j2\pi(M-1)d \sin(\theta_i)/\lambda}]^T$ ($i=1, 2, \dots, N$) 为流形矩阵的第 i 个导向矢量； $\mathbf{n}(t)$ 为 $M \times 1$ 维的噪声矢量，并假定噪声为空间和时间均独立且服从 S α S 分布的复冲击噪声。

常见的低阶矩阵 \mathbf{C} 有如下三种。

(1) 共变矩：共变矩 \mathbf{C}^{ROC} 的元素 C_{ik}^{ROC} 可以根据阵列输出的有限样本数据进行估计，其表达式为

$$C_{ik}^{\text{ROC}} = \frac{\sum_{t=1}^F x_i(t) |x_k(t)|^{p_1-2} x_k^*(t)}{\sum_{t=1}^F |x_k(t)|^{p_1}} \quad (9.1.2)$$

其中, F 为快拍数, p_1 为共变矩参数, 且有 $1 < p_1 < \alpha$, α 代表特征指数。共变矩适用于特征指数大于 1 的情况。

(2) 分数低阶矩: 分数低阶矩 \mathbf{C}^{FLOM} 的元素 C_{ik}^{FLOM} 可以表示为

$$C_{ik}^{\text{FLOM}} = E[|x_i(t)| |x_k(t)|^{p_2-2} x_k^*(t)] \quad (9.1.3)$$

其中, p_2 为分数低阶矩参数, 且 $1 < p_2 < \alpha$ 。此分数低阶矩也仅适用于特征指数大于 1 的情况。

(3) 分数低阶协方差: 分数低阶协方差 \mathbf{C}^{FLOC} 中的元素 C_{ik}^{FLOC} 可以根据阵列输出的有限样本数据由下式进行估计

$$C_{ik}^{\text{FLOC}} = E[|x_i(t)|^{p_3-1} x_i(t) (|x_k(t)|^{p_3-1} x_k(t))^*] \quad (9.1.4)$$

其中, p_3 为分数低阶协方差参数, 且 $0 < p_3 < 1$ 。

9.1.2 三种低阶矩的对比

对低阶矩 \mathbf{C} 进行特征分解得到其信号子空间和噪声子空间, 则

$$\mathbf{C} = \mathbf{U}_s \mathbf{E}_s \mathbf{U}_s^H + \mathbf{U}_n \mathbf{E}_n \mathbf{U}_n^H \quad (9.1.5)$$

其中, \mathbf{U}_s 是由大特征值对应的特征矢量张成的信号子空间, 而 \mathbf{U}_n 是由小特征值对应的特征矢量张成的噪声子空间; \mathbf{E}_s 是由大特征值组成的对角阵, 而 \mathbf{E}_n 是由小特征值组成的对角阵。

基于低阶矩阵的 MUSIC 算法的表达式为

$$P(\theta) = \frac{1}{\mathbf{a}^H(\theta) \mathbf{U}_n \mathbf{U}_n^H \mathbf{a}(\theta)} \quad (9.1.6)$$

式 (9.1.5) 的低阶矩 \mathbf{C} 使用共变矩 \mathbf{C}^{ROC} , 则可得到基于共变矩的 MUSIC 算法, 记作 ROC-MUSIC; 同理, 式 (9.1.5) 的低阶矩 \mathbf{C} 使用分数低阶矩 \mathbf{C}^{FLOM} , 就可得到基于分数低阶矩的 MUSIC 算法, 记作 FLOM-MUSIC; 式 (9.1.5) 的

低阶矩 \mathbf{C} 使用分数低阶协方差 \mathbf{C}^{FLOC} ，就可得到基于分数低阶协方差的 MUSIC 算法，记作 FLOC-MUSIC。

为了验证所提算法的有效性，该节设计了几个试验进行计算机仿真验证所提算法的有效性。阵列由 8 个相距半波长的均匀线阵组成，快拍数取 512。源信号为两个等功率的复高斯过程。由于 S α S 分布不具有二阶矩，则定义如下的广义信噪比 (GSNR)

$$\text{GSNR} = 10 \lg \left[\sum_{i=1}^N \varepsilon_i^2 / (N\gamma) \right] \quad (9.1.7)$$

其中， ε_i^2 为第 i 个高斯信号的功率， γ 为冲击噪声的离差。

在特征指数为 1.6 且广义信噪比为 15dB 时，对来波方向为 $\{0^\circ, 8^\circ\}$ 和 $\{0^\circ, 5^\circ\}$ 的情况分别进行试验仿真比较，仿真结果如图 9.1.1 和和图 9.1.2 所示。从图 9.1.1 中可以看出，虽然三种算法都可以分辨出两个信源，但 FLOC-MUSIC 算法的谱峰更尖锐，效果也更好。而图 9.1.2 表明对于距离较近的两个信源，在 ROC-MUSIC 算法不能分辨的情形下，FLOC-MUSIC 算法可以有效地分辨，FLOM-MUSIC 算法的分辨效果较差。综上所述，可以证明 FLOC-MUSIC 算法具有良好的性能和分辨能力。

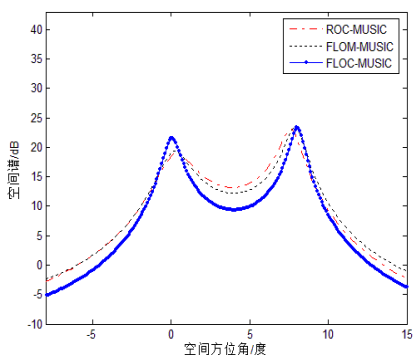


图 9.1.1 8°和 0°独立信源谱峰比较图

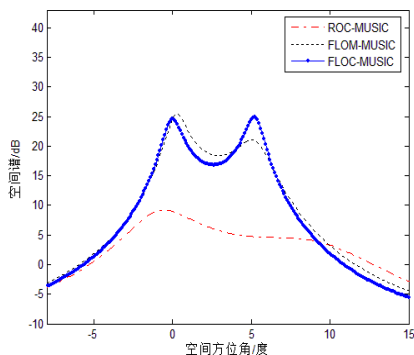


图 9.1.2 5°和 0°独立信源谱峰比较图

9.2 基于量子文化细菌觅食算法的无穷范数最大似然测向方法^[10]

本节提出了量子文化细菌觅食算法 (Quantum-inspired Cultural Bacterial Foraging Algorithm, QCBFA), 用于解决冲击噪声下的测向问题。该算法将量子计算和文化算法应用于细菌觅食优化 (Bacterial Foraging Optimization, BFO) 算法, 增加了潜在解的多样性, 从而降低了计算量并加快了收敛速度。然后将该算法应用于冲击噪声下的测向问题, 并与无穷范数极大似然 (Infinite Norm Maximum Likelihood, INML) 算法相结合, 提出了 QCBFA-INML 算法。Monte Carlo 仿真表明, QCBFA-INML 算法在冲击噪声环境下, 对于相干与非相干信源的测向均有良好的性能。

9.2.1 量子文化细菌觅食算法

假设要寻找 $f(\bar{\mathbf{x}}_q), \bar{\mathbf{x}}_q \in \mathbf{R}_M$ 的最大值, 其中 \mathbf{R}_M 为实数的 M 维空间。假设 $\bar{\mathbf{x}}_q = [\bar{x}_{q1}, \bar{x}_{q2}, \dots, \bar{x}_{qM}]$ 为第 q 个细菌位置, 即量子位置 $\mathbf{x}_q = [x_{q1}, x_{q2}, \dots, x_{qM}]$ 的映射, 其中 $0 \leq x_{qi} \leq 1 (i=1, 2, \dots, M)$ 。第 q 个位置的第 i 维为 $\bar{x}_{qi} = \bar{x}_i^{\min} + x_{qi}(\bar{x}_i^{\max} - \bar{x}_i^{\min})$, 其中 $\bar{x}_{qi} \in [\bar{x}_i^{\min}, \bar{x}_i^{\max}]$; \bar{x}_i^{\min} 为第 i 维变量的下限; \bar{x}_i^{\max} 为第 i 维变量的上限。此外, 细菌 q 的适应度值为 $f(\bar{\mathbf{x}}_q)$ 。

接受函数与影响函数在信仰空间与细菌种群空间之间起到了交流渠道的作用。针对不同的问题, 不同的量子知识来源将对最优位置的产生有不同的影响。在 QCBFA 中, 仅考虑两个知识源, 即量子规范知识和量子形势知识。相应地, 量子信仰空间的结构被定义为 $\{\mathbf{G}_i, \mathbf{s}, i=1, 2, \dots, M\}$, 其中 \mathbf{G}_i 为量子规范知识, 它包含规范知识参数范围的信息; \mathbf{s} 为量子形势知识, 它包含最优量子位置的信息。 \mathbf{G}_i 被定义为 $\langle \mathbf{I}_i, L_i, U_i \rangle$, 其中 $\mathbf{I}_i = [l_i, u_i]$ 为第 i 维量子规范知识的范围区间, 下限 l_i 与上限 u_i 均由给定的量子定义域进行初始化, 且在迭代的过程中由接受

函数进行更新； L_i 为下限 l_i 所对应的适应度值， U_i 为上限 u_i 所对应的适应度值。对于最大值优化问题， L_i 和 U_i 均被初始化为 $-\infty$ 。

接受函数从细菌种群中选择优秀量子位置更新知识源，这将直接影响当前的量子信仰空间。选择最优量子位置来更新量子形势知识 s ，量子形势知识 s 的更新方程为

$$s^{g+1} = \begin{cases} x_{\text{best}}^{g+1}, & \text{if } f(\bar{x}_{\text{best}}^{g+1}) > f(\bar{s}^g) \\ s^g, & \text{else} \end{cases} \quad (9.2.1)$$

其中， $\bar{x}_{\text{best}}^{g+1}$ 为第 $g+1$ 代种群空间中的最优位置； x_{best}^{g+1} 为第 $g+1$ 代种群空间中的最优量子位置； \bar{s}^g 为第 g 代为止所找到的全局最优位置，即量子形势知识 s^g 的映射态。

在接受函数中可选出 20%的较优量子位置，其被用来更新量子规范知识 G_i ，假设第 q 个细菌将影响量子规范知识第 i 个参数的下限，则规范知识 G_i 的更新方程为

$$l_i^{g+1} = \begin{cases} x_{qi}^g, & \text{if } x_{qi}^g \leq l_i^g \text{ or } f(\bar{x}_q^g) > L_i^g \\ l_i^g, & \text{else} \end{cases} \quad (9.2.2)$$

$$L_i^{g+1} = \begin{cases} f(\bar{x}_q^g), & \text{if } x_{qi}^g \leq l_i^g \text{ or } f(\bar{x}_q^g) > L_i^g \\ L_i^g, & \text{else} \end{cases} \quad (9.2.3)$$

其中， l_i^g 为第 g 次迭代第 i 个参数的量子规范知识的下限， L_i^g 为其评价价值。

假设第 q 个细菌将影响量子规范知识第 i 个参数的上限，则其更新方程为

$$u_i^{g+1} = \begin{cases} x_{qi}^g, & \text{if } x_{qi}^g \geq u_i^g \text{ or } f(\bar{x}_q^g) > U_i^g \\ u_i^g, & \text{else} \end{cases} \quad (9.2.4)$$

$$U_i^{g+1} = \begin{cases} f(\bar{x}_q^g), & \text{if } x_{qi}^g \geq u_i^g \text{ or } f(\bar{x}_q^g) > U_i^g \\ U_i^g, & \text{else} \end{cases} \quad (9.2.5)$$

其中， u_i^g 为第 g 次迭代第 i 个参数的量子规范知识的上限； U_i^g 为其评价价值。

经典的 BFO 算法包括趋化行为、聚集行为、复制行为和迁徙行为。趋化行

为是细菌自发地向营养丰富的区域聚集的行为；聚集行为是细菌个体之间通过相互作用进行通信的生物行为；复制行为源于自然选择的概念，只有最适应环境的细菌才能够存活下来，并将其生物特性成功地遗传给下一代，而适应性弱的个体则往往被淘汰；迁徙行为是选择部分细菌将其消除并以一定概率在搜索空间中随机被驱散到（即生成）一个新的量子位置，这能够确保物种的多样性。

从根本上说，趋化行为是细菌觅食行为的关键，细菌试图到达营养集中的区域，避免有毒物质，从而完成优化。这一过程将通过游走的方式来完成，因此可能限制其寻找全局最优解的能力。对于细菌的趋化行为，设计了两个新的量子搜索策略，这将明显提高 BFO 算法的性能。

在 M 维优化问题中， H 个细菌的初始量子位置是随机产生的，第 q 个细菌的量子位置为 $\mathbf{x}_q = [x_{q1}, x_{q2}, \dots, x_{qM}]$ ($q=1, 2, \dots, H$)。将细菌的最优量子位置保存于量子形势知识 $\mathbf{s} = [s_1, s_2, \dots, s_M]$ 中，随后，细菌的量子位置由两种新的趋化行为进行更新。第 q 个细菌的量子位置可由如下的趋化行为进行更新：

$$\delta_{qi}^{j+1} = r_1 \cdot (s_i^j - x_{qi}^j) + N(0, 1) \cdot |z_i^j - x_{qi}^j| \quad (9.2.6)$$

$$x_{qi}^{j+1} = |x_{qi}^j \cos(\delta_{qi}^{j+1}) + \sqrt{1 - (x_{qi}^j)^2} \sin(\delta_{qi}^{j+1})| \quad (9.2.7)$$

其中， $i=1, 2, \dots, M$ ； r_1 为 $[0, 1]$ 之间服从均匀分布的随机数； $N(0, 1)$ 为服从标准正态分布的随机数； δ_{qi}^{j+1} 为第 $j+1$ 次迭代的量子旋转角； \mathbf{z}^j 为 H 个细菌的量子位置的平均值，其值为

$$\mathbf{z}^j = [z_1^j, z_2^j, \dots, z_M^j] \quad (9.2.8)$$

通过改变量子信仰空间中的搜索范围和变化方向，第 q 个细菌的量子位置也可由如下的趋化行为进行更新：

$$\delta_{qi}^{j+1} = \begin{cases} +|N(0, 1) \cdot \text{size}(\mathbf{I}_i^{j+1})|, & x_{qi}^j < l_i^{j+1} \\ -|N(0, 1) \cdot \text{size}(\mathbf{I}_i^{j+1})|, & x_{qi}^j > u_i^{j+1} \\ \eta \cdot N(0, 1) \cdot \text{size}(\mathbf{I}_i^{j+1}), & \text{其他} \end{cases} \quad (9.2.9)$$

$$x_{qi}^{j+1} = |x_{qi}^j \cos(\delta_{qi}^{j+1}) + \sqrt{1 - (x_{qi}^j)^2} \sin(\delta_{qi}^{j+1})| \quad (9.2.10)$$

其中, $i=1,2,\dots,M$; $N(0,1)$ 为服从标准正态分布的随机数; η 为一个常数, 本文中取 0.06; $\text{size}(\mathbf{I}_i^{j+1}) = u_i^{j+1} - l_i^{j+1}$ 为量子信仰空间中第 i 个参数的量子规范知识长度; δ_{qi}^{j+1} 为第 $j+1$ 次迭代中的量子旋转角。

完成 N_c 次趋化行为后, 需要进行复制行为。对于这一最大值优化问题, $f_{\text{health}}^q(j,k,l)$ 的值越大, 细菌越健壮。为了模拟自然界中的复制特性并加快聚集的速度, 所有的细菌将根据其适应度值进行降序排列, 前 $H/2$ 个细菌将无任何突变地分裂成两个子细菌 (为了方便, H 假设为正偶数)。子细菌将继承母细菌的生物特性, 具有与母细菌相同的位置。通过这一选择过程, 剩余 $H/2$ 个不健壮的细菌将消亡。为了简化, 细菌的总量在整个选择过程中保持不变。

为了提高全局搜索能力, 在进行 N_{re} 次复制行为后需要进行 N_{ed} 次迁徙行为, 即选择部分细菌将其消除并以概率 P_{ed} 驱散到搜索空间中的随机量子位置 (这保持了细菌种群数量不变), 这一行为有助于避免算法陷入局部最优。

在完成 N_c 次趋化行为后, 采用类似于 BFO 算法的复制行为, 即适应度低的 $H/2$ 个细菌消亡, 适应度高的 $H/2$ 个细菌分裂。

通过新旧位置的相互比较, 每个细菌将为下一次的趋化行为选择较好的位置。多次进行趋化行为至规定的迭代次数, 最终输出符合要求的最优位置。

9.2.2 量子文化细菌觅食算法的无穷范数极大似然测向

假设在 N 元各向同性阵列 (均匀线阵) 的远场 $\theta_i (i=1,2,\dots,M)$ 处, 有 M 个窄带点源以平面波入射, 入射波长为 λ , 相邻天线间的距离为 d , 则阵列接收的快拍数据为

$$\mathbf{y}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{h}(t) + \mathbf{n}(t) \quad (9.2.11)$$

其中, $\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_N(t)]^T$ 为 $N \times 1$ 维的阵列接收的快拍数据矢量; $\mathbf{h}(t)$ 为 $M \times 1$ 维的信号矢量; $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_M)]$ 为阵列的流形矩阵, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_M]$ 为信源的方向矢量, $\mathbf{a}(\theta_i) = [1, e^{-j2\pi d \sin(\theta_i)/\lambda}, \dots, e^{-j2\pi(N-1)d \sin(\theta_i)/\lambda}]^T$

$(i=1,2,\dots,M)$ 为流形矩阵中的第 i 个导向矢量； $\mathbf{n}(t)=[n_1(t), n_2(t), \dots, n_N(t)]^T$ 为 $N \times 1$ 维的服从 SaS 分布的复冲击噪声矢量。

对接收到的信号进行无穷范数归一化处理，即 $\mathbf{Y}(t) = \mathbf{y}(t) / \max_{1 \leq n \leq N} \{|y_n(t)|\}$ ，其协方差矩阵为

$$\mathbf{C} = E[\mathbf{Y}(t)\mathbf{Y}^H(t)] \quad (9.2.12)$$

INML 的测向问题实质上是 $\boldsymbol{\theta}=[\theta_1, \theta_2, \dots, \theta_M]$ 的参数估计问题，则其极大似然估计值为

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \text{trace}[\mathbf{P}_{A(\boldsymbol{\theta})}\mathbf{C}] \quad (9.2.13)$$

其中， $\mathbf{P}_{A(\boldsymbol{\theta})} = \mathbf{A}(\boldsymbol{\theta})(\mathbf{A}^H(\boldsymbol{\theta})\mathbf{A}(\boldsymbol{\theta}))^{-1}\mathbf{A}^H(\boldsymbol{\theta})$ 为阵列流形矩阵 $\mathbf{A}(\boldsymbol{\theta})$ 的投影矩阵。

在 QCBFA 中，量子位置的初始量子比特在 $[0,1]$ 之间随机产生。对于冲击噪声下 INML 的测向问题，细菌量子位置所对应的位置代表了测向问题中的一个潜在解。在 INML 的最大值优化问题中，估计参数 $\boldsymbol{\theta}$ 代表细菌的位置，则细菌的适应度函数为

$$f(\boldsymbol{\theta}) = \text{trace}[\mathbf{P}_{A(\boldsymbol{\theta})}\mathbf{C}] \quad (9.2.14)$$

适应度值是用来评估每个细菌状态的。在该测向问题中，其目的是搜寻最大适应度值所对应的细菌位置。细菌的最优位置等价于最优方向矢量 $\boldsymbol{\theta}$ 。因此，测向问题就转化为了搜寻最优位置的连续优化问题。

根据以上介绍，基于 QCBFA-INML 的测向方法描述如下。

Step 01 初始化参数 H 、 N_c 、 N_s 、 N_{re} 、 N_{ed} 、 P_{ed} 、 \mathbf{x}_q ($q=1,2,\dots,H$) 及量子信仰空间。

Step 02 迁徙行为的循环： $l=l+1$ ， $k=0$ 。

Step 03 复制行为的循环： $k=k+1$ ， $j=0$ 。

Step 04 趋化行为的循环： $j=j+1$ 。

(1) 将细菌 q 的当前量子位置保存于 \mathbf{x}_q^j ($q=1,2,\dots,H$) 中。

(2) 映射得到量子位置 \mathbf{x}_q 对应的位置, 并计算其适应度值 $f^q(j, k, l)$ 。

(3) 更新量子信仰空间, 寻找当前最大适应度值, 并将相应的量子位置保存于 \mathbf{s}^j 中。

(4) 翻转: 计算 H 个细菌量子位置的平均值 $\mathbf{z}^j = [z_1^j, z_2^j, \dots, z_M^j]$ 。

(5) 依次对细菌 q 进行趋化, 先令 $q=1$ 。

(6) 通过第一种趋化行为前进, 搜寻较优的适应度值, 并将其保存为 $f_{\text{last}}^q = f^q(j, k, l)$ 。

(7) 移动: 根据适应度值, 更新细菌的量子位置。

(8) 利用 (2) 中的方法计算量子位置 \mathbf{x}_q^{j+1} 的适应度值 $f^q(j+1, k, l)$ 。

(9) 游弋:

a. 令 $m=0$ 。

b. $m=m+1$: 若 $f^q(j+1, k, l) > f_{\text{last}}^q$, 则 $f_{\text{last}}^q = f^q(j+1, k, l)$, $\mathbf{x}_q^j = \mathbf{x}_q^{j+1}$ 。

c. 即细菌 q 随机选择一种趋化行为方式来更新量子位置 $\mathbf{x}_q^{j+1} = [x_{q1}^{j+1}, x_{q2}^{j+1}, \dots, x_{qM}^{j+1}]$, 并利用该量子位置的相应位置计算新的适应度值 $f^q(j+1, k, l)$ 。

d. 如果 $m < N_s$ (N_s 为游弋的最大次数), 返回 b; 否则, $m=N_s$, 本循环结束, 设置 $\mathbf{x}_q^{j+1} = \mathbf{x}_q^j$ 。

(10) 若 $q \neq H$, 则令 $q=q+1$, 返回 (6)。

Step 05 若 $j < N_c$, 则返回 **Step 04** (由于细菌的生命没有结束, 需要继续完成趋化行为)。

Step 06 复制行为:

(1) 根据给定的 k 与 l , 计算细菌 q 的适应度值 $f_{\text{health}}^q = f^q(N_c, k, l)$, 并根据适应度值将细菌降序排列。

(2) 适应度值较小的 $H/2$ 个细菌将消亡, 剩余的 $H/2$ 个细菌将分裂成两个子细菌 (母细菌与子细菌的量子位置相同)。

Step 07 若 $k < N_{re}$, 则返回 **Step 03**。

Step 08 迁徙行为: 除最优个体外, 每个细菌均以概率 P_{ed} 消亡并驱散。若 $l < N_{ed}$, 则返回 **Step 02**; 否则, 输出最优量子位置及其适应度值, 算法结束。

综上所述, 该算法将量子计算与文化算法应用于细菌觅食优化算法, 很好地融合了以上算法的优点, 增加了潜在解的多样性, 从而加快了收敛速度并提高了收敛精度。

此外, 为了比较算法的计算复杂度, 在 QCBFA 中, 当前的迭代次数可以表示为 $g = N_c \cdot (k-1) + N_c \cdot N_{re} \cdot (l-1) + j$ 。

9.2.3 试验仿真

假设冲击噪声服从特征指数为 α 的标准 SaS 分布, 该测向系统由阵元数为 8 的均匀线阵构成, 且相邻阵元的间距均为半波长 ($d = \lambda/2$), 快拍数假设为 200, 所有测向算法均在相同的条件下仿真。

基于 INML 方法进行一系列的仿真试验来对 QCBFA 的相关性能进行评价, 并与经典的智能计算方法进行比较, 如 CA、SFLA 和 FA。这 4 种算法分别称为 CA-INML、SFLA-INML、FA-INML 和 QCBFA-INML。

在 QCBFA 中, 主要参数设置如下: $N_c=50$, $N_s=4$, $N_{re}=1$, $N_{ed}=2$, $P_{ed}=0.1$ 。在 CA 中, 参数的设置参考文献[11]。所有算法的种群中个体数量设置为 100, 终止迭代次数为 100, Monte Carlo 试验次数为 200, 所有仿真结果均为 200 次试验的平均值, 200 次试验的结果用来检验这些智能算法的性能。SFLA 算法^[12]的其他参数设置如下: 族群的数量为 20, 每个族群青蛙的数量为 $n=5$, 一次混排后每个族群内的最大迭代次数为 5。在 FA 中, 烟花的数量为 10, 控制参数 $m=90$, $a=0.04$, $b=0.8$, $\hat{m}=10$, $\xi=10^{-7}$, 其他参数的设置参考文献[13]。

首先考虑统计独立的信号源 $\{\theta_1=50^\circ, \theta_2=30^\circ, \theta_3=11^\circ, \theta_4=1.1^\circ, \theta_5=-20^\circ\}$ 和 $\{\theta_1=50^\circ, \theta_2=30^\circ, \theta_3=11^\circ, \theta_4=1.1^\circ\}$ 。图 9.2.1 和图 9.2.2 给出了在 GSNR 为 10dB 且特征指数为 1.2 时, 对于不同数量的独立信号源 4 种算法的性能。从

图 9.2.1 和图 9.2.2 中可以看出, 对于独立信号源, QCBFA-INML 算法优于 CA-INML、SFLA-INML 和 FA-INML 算法, FA-INML 和 SFLA-INML 算法均陷入了局部最优, 而 CA-INML 算法优于它们, 但劣于 QCBFA-INML 算法。从图 9.2.1 和图 9.2.2 中还可以看出, 对于不同数量的独立信号源, QCBFA-INML 算法比其他三种算法的收敛精度更高。

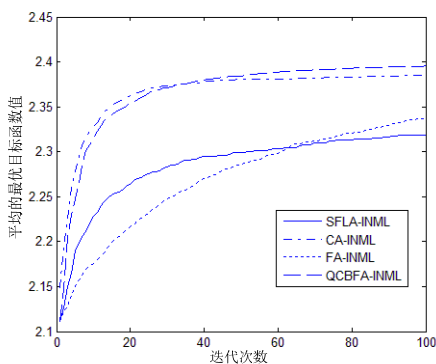


图 9.2.1 对于 5 个独立信号源的
4 种算法的收敛曲线

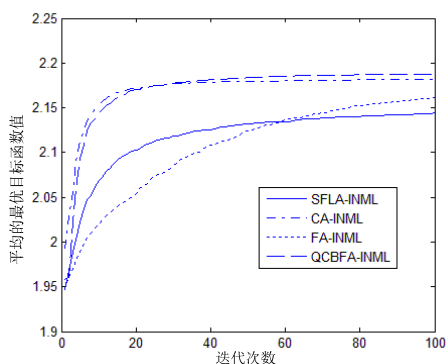


图 9.2.2 对于 4 个独立信号源的
4 种算法的收敛曲线

其次考虑相干的信号源 $\{\theta_1 = 50^\circ, \theta_2 = 30^\circ, \theta_3 = 11^\circ, \theta_4 = 1.1^\circ, \theta_5 = -20^\circ\}$ 和 $\{\theta_1 = 50^\circ, \theta_2 = 30^\circ, \theta_3 = 11^\circ, \theta_4 = 1.1^\circ\}$ 。图 9.2.3 和图 9.2.4 表明了, 在 GSNR 为 10dB 且特征指数为 1.2 时, QCBFA-INML 算法的收敛性优于 CA-INML、SFLA-INML 和 FA-INML 算法。正是由于 QCBFA 算法利用了量子信息和文化信息, 才导致其大大地提高了收敛速度和收敛精度。从图 9.2.3 和图 9.2.4 中可以得出, 对于不同数量的相干信号源, 在 4 种算法中 QCBFO 算法最有效且最稳定。

然后将 QCBFA-INML 算法与已有的测向算法进行比较, 如 FLOM-MUSIC、CA-ROC-ML、CSFL-FLOC-WSSF 算法等, 其中 FLOM-MUSIC 算法为基于 FLOM 的空间平滑 MUSIC 算法 (平滑数量设置为 3), CA-ROC-ML 算法为基于 ROC 和 CA 的 ML 算法, CSFL-FLOC-WSSF 算法为基于 FLOC 和 CSFL 的 WSSF 算法。CSFL 算法^[14]的参数设置如下: 族群的数量为 20, 每个族群青蛙的数量为 $n=5$, 混合前每个族群的迭代次数为 5, 混合的迭代次数为 $Gen=100$ 。CA 和 QCBFA 算法中其他参数的设置与之前的试验相同。

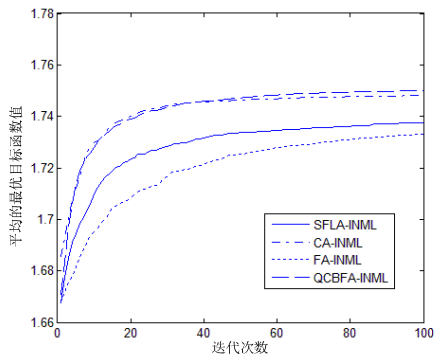


图 9.2.3 对于 5 个相干信号源的 4 种算法的收敛曲线

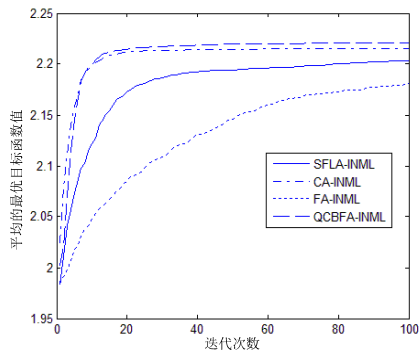


图 9.2.4 对于 4 个相干信号源的 4 种算法的收敛曲线

接着考虑两个统计独立的信号源 $\{\theta_1 = 20^\circ, \theta_2 = 10^\circ\}$ 。图 9.2.5 表明了在不同 GSNR 以 3dB 为间隔从 -5dB 变化到 10dB 时 4 种算法的均方根误差(Root Mean Square Error, RMSE)的对比。从图 9.2.5 中可以看出, QCBFA-INML 算法比其他三种测向算法具有更小的均方根误差。图 9.2.6 表明了在不同特征指数以 0.2 为间隔从 1 变化到 2 时 4 种算法的成功概率曲线。当其估计误差的绝对值小于 1° 时, 就认为估计成功, 成功概率由成功估计的比例计算得出。图 9.2.6 所示的成功概率曲线表明, 在环境变化的情况下, QCBFA-INML 算法既有效又可靠。图 9.2.5 和图 9.2.6 均表明, 在冲击噪声下, 对于独立信号源, QCBFA-INML 算法优于 FLOM-MUSIC、CA-ROC-ML 和 CSFL-FLOC-WSSF 算法。

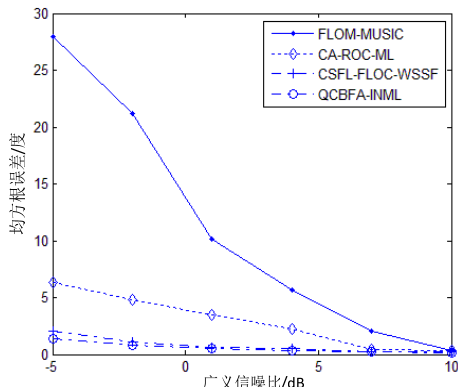


图 9.2.5 特征指数为 1.6 时独立信号源的均方根误差

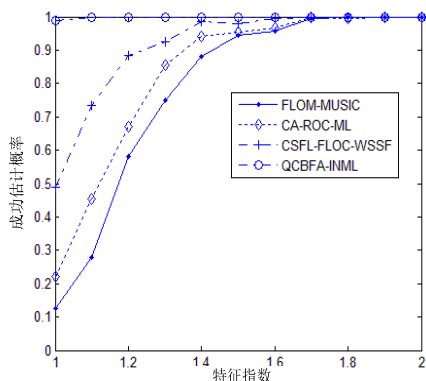


图 9.2.6 GSNR 为 10dB 时独立信号源的成功概率

再考虑两个不同的统计独立的信号源 $\{\theta_1 = 30^\circ, \theta_2 = 18^\circ\}$ 。图 9.2.7 表明了特征指数为 1.6 且 GSNR 以 3dB 为间隔从 -5dB 变化到 10dB 时 4 种算法的 RMSE 的对比。图 9.2.8 表明了 GSNR 为 10dB 且特征指数以 0.2 为间隔从 1 变化到 2 时 4 种算法的成功概率曲线。图 9.2.7 和图 9.2.8 均表明, 在冲击噪声下, 对于独立信号源, QCBFA-INML 算法优于 FLOM-MUSIC、CA-ROC-ML 和 CSFL-FLOC-WSSF 算法。

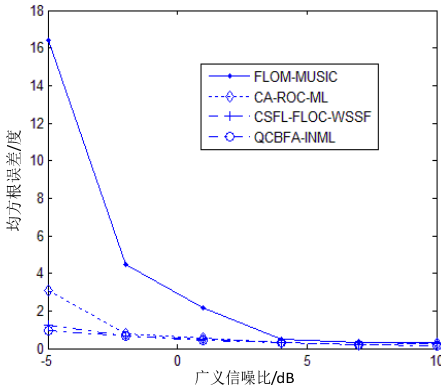


图 9.2.7 特征指数为 1.6 时独立信号源的均方根误差

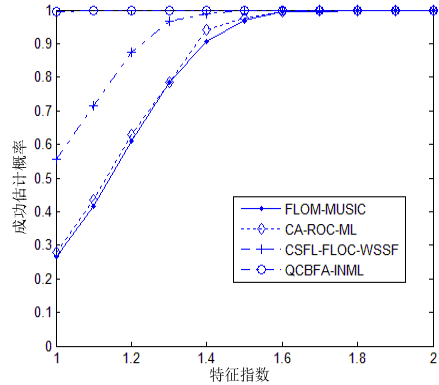


图 9.2.8 GSNR 为 10dB 时独立信号源的成功概率

从图 9.2.5 和图 9.2.7 中可以看出独立信号源间不同角度差的 RMSE 的对比。由于图 9.2.7 中的角度差较大, 则其 RMSE 较小。此外, 对于不同角度差的独立信号源, QCBFA-INML 算法比其他三种算法的估计精度更高。而图 9.2.6 和图 9.2.8 表现了独立信号源间不同角度差的成功概率的对比, 其角度差越大, 成功概率越高, 并且 QCBFA-INML 算法更加有效且稳定。

最后考虑两个相干的信号源 $\{\theta_1 = 5^\circ, \theta_2 = 25^\circ\}$ 。图 9.2.9 表明了特征指数为 1.8 且 GSNR 以 3dB 为间隔从 -5dB 变化到 10dB 时 4 种算法的 RMSE 的对比。图 9.2.10 表明了 GSNR 为 10dB 且特征指数以 0.2 为间隔从 1 变化到 2 时 4 种算法的成功概率曲线。图 9.2.9 和图 9.2.10 均表明, QCBFA-INML 算法优于 FLOM-SSMUSIC、CA-ROC-ML 和 CSFL-FLOC-WSSF 算法。这是由于 QCBFA-INML 算法充分利用了无穷范数和 QCBFA, 因此能够抵抗冲击噪声并提高 ML 算法的性能。

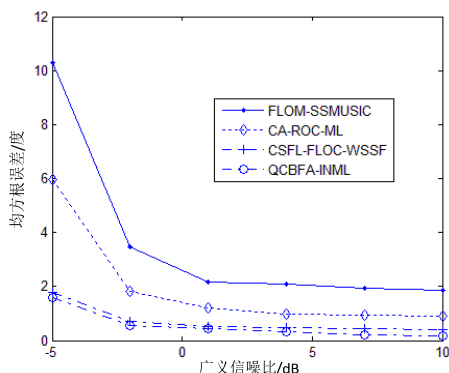


图 9.2.9 特征指数为 1.8 时相干信号源的均方根误差对比曲线

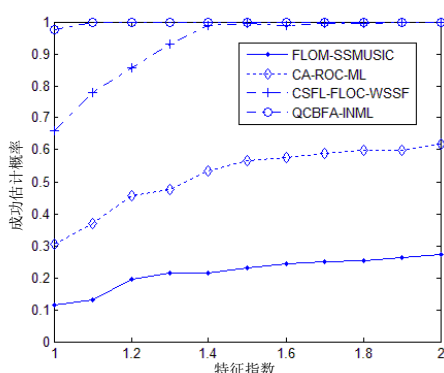


图 9.2.10 GSNR 为 10dB 时相干信号源的成功概率对比曲线

9.3 基于量子粒子群的动态测向方法

目前，高斯环境下的动态测向问题已被广泛研究，设计出了一些有效的动态测向方法^[15-17]，但冲击噪声下的动态目标的鲁棒测向问题仍旧是一个理论难题。

为了解决冲击噪声下的动态目标测向问题，本节提出了 WSC-ML 算法，然后利用量子粒子群优化（Quantum Particle Swarm Optimization, QPSO）算法求解基于 WSC-ML 算法的跟踪方程。该算法（QPSO-WSC-ML）避免了基于子空间的动态测向算法中分数低阶矩阵的抗冲击能力不强的问题，在精度和鲁棒性上均有明显优势。

9.3.1 冲击噪声下的动态测向模型

假设在 N 元各向同性阵列（均匀线阵）的远场处，有 M 个窄带点源以 $\theta_i (i=1,2,\dots,M)$ 方向平面波入射，入射波长为 λ ，相邻天线间的距离 d ，则阵列接收的快拍数据为

$$\mathbf{x}(k) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(k) + \mathbf{n}(k) \quad (9.3.1)$$

其中, $\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_N(k)]^T$ 为 $N \times 1$ 维的阵列接收的快拍数据矢量; $\mathbf{s}(k)$ 为 $M \times 1$ 维信号矢量; $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \dots, \mathbf{a}(\theta_M)]$ 为阵列的流形矩阵, $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_M]$ 为信源的方向矢量, $\mathbf{a}(\theta_i) = [1, e^{-j2\pi d \sin(\theta_i)/\lambda}, \dots, e^{-j2\pi(N-1)d \sin(\theta_i)/\lambda}]^T$ ($i=1, 2, \dots, M$) 为流形矩阵的第 i 个导向矢量; $\mathbf{n}(k)$ 为 $N \times 1$ 维的服从 SaS 分布的复冲击噪声矢量。

由于冲击噪声环境下, 接收的快拍数据的二阶及以上矩不存在, 则定义加权的接收快拍数据矢量为 $\bar{\mathbf{x}}(k) = \mathbf{x}(k) / \left(\max_{1 \leq i \leq N} |x_i(k)|^\sigma \right)$, 其中 σ 为常数。因此, 接收第 1 个快拍数据的加权信号协方差矩阵为

$$\mathbf{C}(1) = \bar{\mathbf{x}}(1)\bar{\mathbf{x}}^H(1) \quad (9.3.2)$$

而接收第 $(k+1)$ 个快拍数据的加权信号协方差矩阵的更新方程为

$$\mathbf{C}(k+1) = \mu \mathbf{C}(k) + (1-\mu) \bar{\mathbf{C}}(k+1) \quad (9.3.3)$$

其中, $\mathbf{C}(k)$ 为 k 个快拍数据的加权信号协方差矩阵; $\bar{\mathbf{C}}(k+1) = \bar{\mathbf{x}}(k+1)\bar{\mathbf{x}}^H(k+1)$ 为第 $(k+1)$ 个加权的接收快拍数据的协方差矩阵; μ 为更新因子。

动态测向问题实质上是 $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_M]$ 的参数估计问题, 则信号源的极大似然估计值为

$$\bar{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} [\text{trace}(\mathbf{P}_{\mathbf{A}(\boldsymbol{\theta})} \mathbf{C}(k))] \quad (9.3.4)$$

其中, $\mathbf{P}_{\mathbf{A}(\boldsymbol{\theta})} = \mathbf{A}(\boldsymbol{\theta})(\mathbf{A}^H(\boldsymbol{\theta})\mathbf{A}(\boldsymbol{\theta}))^{-1}\mathbf{A}^H(\boldsymbol{\theta})$ 为阵列流形矩阵 $\mathbf{A}(\boldsymbol{\theta})$ 的投影矩阵; $\mathbf{C}(k)$ 为第 k 个快拍数据的加权信号协方差矩阵。

这里可以将 WSC-ML 算法看作一个典型的连续优化问题, 利用连续的 QPSO 算法可以寻求其全局最优解, 即可得到移动信号源的方向。

9.3.2 连续量子粒子群优化算法

量子粒子群优化算法是一种由粒子群的量子行为所启发的新型智能优化算

法，其中量子位置用来产生相应的粒子位置，而粒子位置则代表一个潜在解，是量子位置在定义域的映射态。第 q 个粒子的量子位置被定义为

$$\mathbf{y}_q = [\mathbf{y}_{q1}, \mathbf{y}_{q2}, \dots, \mathbf{y}_{qM}] = \begin{bmatrix} \alpha_{q1}, \alpha_{q2}, \dots, \alpha_{qM} \\ \beta_{q1}, \beta_{q2}, \dots, \beta_{qM} \end{bmatrix} \quad (9.3.5)$$

其中， α_{qm} 和 β_{qm} 为一对量子比特位置，且满足归一化条件 $|\alpha_{qm}|^2 + |\beta_{qm}|^2 = 1$ ， $0 \leq \alpha_{qm} \leq 1, 0 \leq \beta_{qm} \leq 1, m = 1, 2, \dots, M$ 。

假设量子粒子群中有 Q 个 M 维粒子，其中第 q 个粒子的量子位置为 $\mathbf{y}_q = [\mathbf{y}_{q1}, \mathbf{y}_{q2}, \dots, \mathbf{y}_{qM}]$ ($q = 1, 2, \dots, Q$)，粒子速度为 $\mathbf{v}_q = [\mathbf{v}_{q1}, \mathbf{v}_{q2}, \dots, \mathbf{v}_{qM}]$ ， $\mathbf{p}_q = [\mathbf{p}_{q1}, \mathbf{p}_{q2}, \dots, \mathbf{p}_{qM}]$ 和 $\mathbf{p}_g = [\mathbf{p}_{g1}, \mathbf{p}_{g2}, \dots, \mathbf{p}_{gM}]$ 分别代表第 q 个粒子的局部最优量子位置和整个种群全局最优量子位置，则第 q 个粒子速度的更新方程为

$$\mathbf{v}_{qm}^{t+1} = w\mathbf{v}_{qm}^t + c_1r_1(\mathbf{p}_{qm}^t - \beta_{qm}^t) + c_2r_2(\mathbf{p}_{gm}^t - \beta_{qm}^t) \quad (9.3.6)$$

其中， c_1 和 c_2 为控制最大搜索步长的加速系数； r_1 和 r_2 为 $[0, 1]$ 间的均匀随机数； w 为 $[0.4, 0.9]$ 间的加权系数，它可以表示为

$$w = w_{\max} - t \times (w_{\max} - w_{\min}) / t_{\max} \quad (9.3.7)$$

其中， w_{\max} 和 w_{\min} 分别为最大加权系数和最小加权系数； t 和 t_{\max} 分别为当前迭代次数和最大迭代次数。

另外，为了防止速度 \mathbf{v}_{qm}^{t+1} 过大使所处位置超过搜索边界，需要通过下式完成速度限制。

$$\mathbf{v}_{qm}^{t+1} = \begin{cases} \mathbf{v}_{m,\max}, & \mathbf{v}_{qm}^{t+1} > \mathbf{v}_{m,\max} \\ -\mathbf{v}_{m,\max}, & \mathbf{v}_{qm}^{t+1} < -\mathbf{v}_{m,\max} \end{cases} \quad (9.3.8)$$

其中， $\mathbf{v}_{m,\max}$ 为第 m 维的最大速度限制。

量子位置的更新过程主要通过量子旋转门来完成，则其更新方程为

$$\mathbf{y}_{qm}^{t+1} = \text{abs}[U(\mathbf{v}_{qm}^{t+1}) \mathbf{y}_{qm}^t] = \text{abs} \left(\begin{bmatrix} \cos(\mathbf{v}_{qm}^{t+1}) & -\sin(\mathbf{v}_{qm}^{t+1}) \\ \sin(\mathbf{v}_{qm}^{t+1}) & \cos(\mathbf{v}_{qm}^{t+1}) \end{bmatrix} \mathbf{y}_{qm}^t \right) \quad (9.3.9)$$

其中， \mathbf{v}_{qm}^{t+1} 为由速度确定的量子旋转角。

9.3.3 量子粒子群优化算法的动态测向方法^[18]

在 QPSO 中, 初始搜索范围设置为信号源方向角的定义域, 当目标移动时, 搜索范围将随着之前的估计值和相关参数而改变, 其搜索范围为

$$\begin{cases} z_m^{\min}(k+1) = z_m^c(k+1) - \varepsilon^k |z_m^{\min}(k) - \bar{z}_m(k)| - r \\ z_m^{\max}(k+1) = z_m^c(k+1) + \varepsilon^k |z_m^{\max}(k) - \bar{z}_m(k)| + r \end{cases} \quad (9.3.10)$$

其中, ε 为搜索空间中影响收敛速率的收敛因子; r 为搜索空间的搜索半径; $z_m^{\max}(k)$ 和 $z_m^{\min}(k)$ 分别为第 k 个快拍第 m 维的搜索上限和搜索下限; $\bar{z}_m(k)$ 为第 k 个快拍第 m 维的估计值; $z_m^c(k+1)$ 为第 $(k+1)$ 个快拍第 m 维的中心值, 即

$$z_m^c(k+1) = \delta z_m^c(k) + (1-\delta)\bar{z}_m(k) \quad (9.3.11)$$

其中, δ 为遗传因子。

假设第 q 个粒子位置 $\mathbf{z}_q = [z_{q1}, z_{q2}, \dots, z_{qM}]$ 为量子位置 $\bar{\mathbf{y}}_q = [\beta_{q1}, \beta_{q2}, \dots, \beta_{qM}]$ 的相应映射态, 其中 $z_{qm} = z_m^{\min} + \beta_{qm}(z_m^{\max} - z_m^{\min})$, $z_{qm} \in [z_m^{\min}, z_m^{\max}]$, z_m^{\max} 和 z_m^{\min} 分别为粒子位置第 m 维的上限和下限。

量子粒子群算法的初始种群在可行域内随机产生, 并用适应度函数来评估每个粒子的状态。对于该跟踪方法, 其适应度函数为

$$f(\mathbf{z}_q) = \text{trace}[\mathbf{P}_{\mathcal{A}(\mathbf{z}_q)} \mathbf{C}(k)] \quad (9.3.12)$$

根据以上介绍与分析, 动态测向方法 QPSO-WSC-ML 可以描述如下。

Step 01 根据第一个快拍采样数据, 初始化加权信号协方差矩阵 $\mathbf{C}(1)$, 其中 $\mathbf{C}(1) = \bar{\mathbf{x}}(1)\bar{\mathbf{x}}^H(1)$, 根据变量的定义域初始化搜索范围。

Step 02 通过随机方法初始化量子位置和速度。

Step 03 计算每个粒子的量子位置的适应度值, 并初始化局部最优量子位置和全局最优量子位置; 第 k 个快拍测向的迭代次数为 $t_{\max} = H \cdot \text{round}[z_1^{\max}(k) - z_1^{\min}(k)]$, 其中 H 为常数; $\text{round}()$ 为取整函数。

Step 04 更新每个粒子的速度和量子位置。

Step 05 根据相应的量子位置的映射态，计算相应的适应度值。

Step 06 更新局部最优量子位置和全局最优量子位置。

Step 07 判断是否达到迭代次数。如果未达到，则 $t = t + 1$ ，返回 **Step 04** 继续迭代；否则，输出全局最优位置并进入下一步。

Step 08 更新加权信号协方差矩阵和搜索范围。

Step 09 判断是否达到最大快拍采样次数。如果未达到，则 $k = k + 1$ ，返回

Step 02 继续迭代；否则，结束迭代。

9.3.4 试验仿真

仿真中一些参数的设置如下：该动态测向系统由阵元数为 8 的均匀线阵构成，且相邻阵元的间距均为半波长，快拍数为 500，假设冲击噪声服从特征指数为 α 的标准 SaS 分布，广义信噪比为 $\text{GSNR} = 10 \lg[E(|s(k)|^2)/\gamma]$ 。其中， $E(|s(k)|^2)$ 为信号的平均功率； γ 为冲击噪声的离差。

接下来，为了表明 QPSO-WSC-ML 算法的优势，将其与 PSO-FLOM-ML 算法^[19]进行比较，其中一些参数的设置如下： $\varepsilon = 0.995$ ， $\mu = 0.95$ ， $r = 3$ ， $\delta = 0.8$ ， $H = 6$ ， $\sigma = 0.7$ ， $w_{\max} = 0.9$ ， $w_{\min} = 0.4$ ， $v_{m,\max} = 0.2$ ， $c_1 = c_2 = 1$ ，种群中粒子数为 20。

首先，考虑两个独立的信号源 $\theta_1(k) = [-12 - 0.01k]^\circ$ 和 $\theta_2(k) = [28 + 0.01k]^\circ$ 。图 9.3.1 和图 9.3.2 给出了两种动态测向方法在 $\alpha = 1.85$ 且 $\text{GSNR} = 4\text{dB}$ 时方向跟踪的性能。从图中可以看出，QPSO-WSC-ML 算法的鲁棒性明显优于 PSO-FLOM-ML 算法。

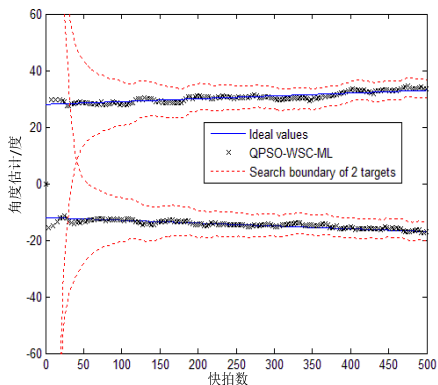


图 9.3.1 基于 QPSO-WSC-ML 算法的两个独立信号源的方向跟踪

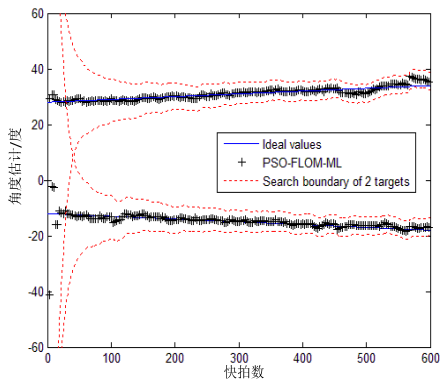


图 9.3.2 基于 PSO-FLOM-ML 算法的两个独立信号源的方向跟踪

其次，考虑三个独立的信号源 $\theta_1(k) = [-30 + 5 \sin(2\pi k / 500)]^\circ$ 、 $\theta_2(k) = [13 + 5 \sin(2\pi k / 500)]^\circ$ 和 $\theta_3(k) = [48 + 5 \sin(2\pi k / 500)]^\circ$ 。图 9.3.3 和图 9.3.4 给出了两种动态测向算法在 $\alpha = 1.5$ 且 $\text{GSNR} = 10\text{dB}$ 时的测向性能。从图中可以看出，QPSO-WSC-ML 算法的跟踪性能明显优于 PSO-FLOM-ML 算法。

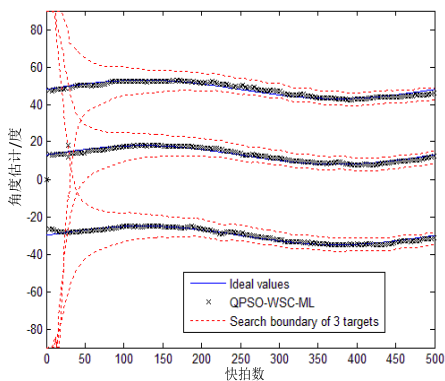


图 9.3.3 基于 QPSO-WSC-ML 算法的三个独立信号源的方向跟踪

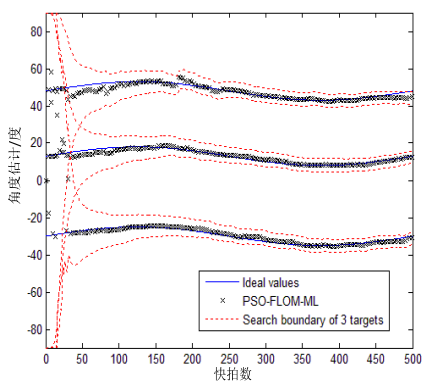


图 9.3.4 基于 PSO-FLOM-ML 算法的三个独立信号源的方向跟踪

最后，图 9.3.5 和图 9.3.6 给出了两种动态测向方法在 $\alpha = 1.5$ 且 $\text{GSNR} = 10\text{dB}$ 时对于三个相干信号源的方向测向性能。从图中可以看出，与 PSO-FLOM-ML 算法相比，QPSO-WSC-ML 算法更具有鲁棒性。

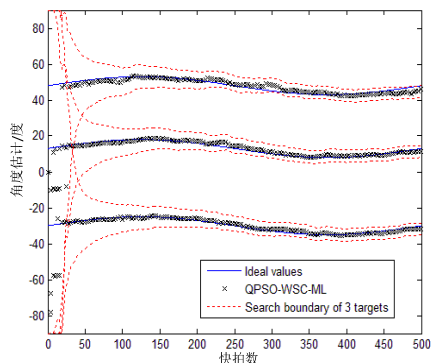


图 9.3.5 基于 QPSO-WSC-ML 算法的三个相干信号源的方向跟踪

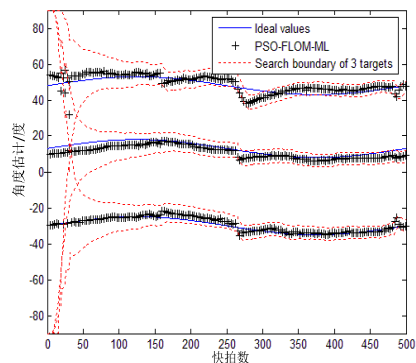


图 9.3.6 基于 PSO-FLOM-ML 算法的三个相干信号源的方向跟踪

从图 9.3.3~图 9.3.6 可知，在相同的仿真条件下，独立信号源的动态测向较相干信号源的动态测向容易。

9.4 小结

本章提出的量子文化细菌觅食算法(QCBFA)是一种新型的连续优化算法，主要设计了趋化行为的两个新的搜索策略，并将其应用于冲击噪声测向模型，从而提出了 QCBFA-INML 测向方法。仿真结果表明，QCBFA 算法在搜索速度和搜索精度上均明显优于 FA、CA 和 SFLA 算法；QCBFA-INML 算法具有良好的性能，并胜过一些经典的测向方法，在冲击噪声下对相干及非相干信号源都能够进行有效的测向。然后，基于 WSC-ML 和 QPSO 算法，提出了一种在冲击噪声下动态目标的方向跟踪方法(QPSO-WSC-ML)。QPSO 算法解决了动态测向这一连续优化问题，减少了 WSC-ML 算法的计算量并提高了 PSO 算法的搜索精度。

参考文献

- [1] Tsihrintzis G A, Nikias C L. Fast estimation of the parameters of alpha-stable impulsive interference. IEEE Trans Signal Process, 1996, 6(44): 1492-1503.
- [2] Hari K V S and Lalitha V. Subspace-based DOA estimation using fractional lower order statistics. Proc. IEEE International Conf. on Acoustics, Speech and Signal Processing, 2011: 2580-2583.
- [3] Liu T H, Mendel J M. A subspace based direction finding algorithm using fractional lower order statistics. IEEE Transactions on Signal Processing, 2001, 49(8): 1605-1613.
- [4] 高洪元, 刁鸣. 重构分数低阶协方差的子空间拟合测向算法. 电波科学学报, 2009, 24 (4): 729-734.
- [5] Zhao Dayong, Gao Hongyuan, Diao Ming, An Chunlian. Direction finding of maximum likelihood algorithm using artificial bee colony in the impulsive noise. The 2010 International Conference on Artificial Intelligence and Computational Intelligence, 2010, 2: 102-105.
- [6] 高洪元, 刁鸣. 基于文化粒子群算法的广义 OPM 测向. 电子信息对抗技术, 2010, 25 (3): 1-5.
- [7] 夏铁骑, 万群, 汪学刚. 冲击噪声环境下基于任意阵列流形的空时二维 DOA 估计方法. 航空学报, 2008, 29 (5): 1233-1238.
- [8] Diao Ming, Li Li, Gao Hongyuan. DOA tracking based on MIMO radar in impulsive noise background. Signal Processing(ICSP), 2015.
- [9] 何劲, 刘中. 利用分数低阶空时矩阵进行冲击噪声环境下的 DOA 估计. 航空学报, 2006, 27 (1): 104-108.

- [10] Gao Hongyuan, Xu Chongqiang and Li Chenwan. Quantum-inspired cultural bacterial foraging algorithm for direction finding of impulse noise. *International Journal of Innovative Computing and Applications*, 2014, 6(1): 44-54.
- [11] 刁鸣, 李晓刚, 王冰. 文化算法的最大似然测向方法研究. *哈尔滨工程大学学报*, 2008, 29 (5): 509-513.
- [12] Eusuff M M and Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 2003, 129(3): 210-225.
- [13] Tan Y and Zhu Y. *Fireworks algorithm for optimization*. *Advances in Swarm Intelligence*, Springer Berlin Heidelberg, 2010: 355-364.
- [14] Gao Hongyuan, Zhao Zhongkai, Wang Wei. Direction finding of weighted signal subspace fitting based on cultural shuffled frog leaping. *Journal of Computational Information Systems*, 2010, 6(3): 847-854.
- [15] 赵大勇, 袁熹, 高洪元. 应用粒子群算法的动态目标 DOA 估计. *哈尔滨工程大学学报*, 2009, 30 (7): 843-846.
- [16] 陈辉, 王永良. 秩-1 子空间跟踪算法. *电子与信息学报*, 2002, 24 (5): 626-630.
- [17] 刁鸣, 袁熹, 高洪元. 一种新的基于粒子群算法的 DOA 跟踪方法. *系统工程与电子技术*, 2009, 31 (9): 2046-2049.
- [18] Gao Hongyuan, Li Jia, Du Yanan. Direction Tracking of multiple moving targets using quantum particle swarm optimization. *International Conference on ISFST*, 2016.
- [19] 赵大勇, 刁鸣, 杨丽丽, 等. 冲击噪声背景下的动态 DOA 跟踪. *山东大学学报 (工学版)*, 2010, 40 (1): 133-138.

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E - m a i l: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036